DRIVES

NOVOS HORIZONTES PARA SEU MSX

CARVALHO JR. — OLIVEIRA — PIAZZI

CARVALHO JR. - OLIVEIRA - PIAZZI

DRIVES

NOVOS HORIZONTES PARA SEU MSX

Coordenação Editorial : Pierluigi Piazzi
Coordenação Didática : Betty F. Piazzi
Produção Editorial : Rosa K. Fromer
Editoração : Renato da Silva Oliveira
Produção do software: Luiz Tarcísio de Carvalho Jr.
Ilustrações e Arte : Ana Lúcia Antico

ALEPH Publicações e Assessoria Pedagógica Ltda.
CP 20.707 CEP 01498 S. Paulo
Tel: (011) 843-3202

SUMÁRIO

INTRODUÇÃO

Este livro se destina ao leitor que deseja se informar sobre os novos horizontes que seu MSX lhe oferece ao conectá-lo a um acionador de disco (drive) ou àquele que, já possuindo um destes periféricos, deseje tirar o máximo proveito de seu equipamento.

Ao escrevê-lo, partimos do pressuposto que o leitor tenha alguma familiaridade com o micro MSX, mas que nunca tenha operado um periférico do tipo citado.

Se o leitor desejar informações adicionais sobre o MSX, sua arquitetura e sua linguagem residente, sugerimos consultar, no fim deste livro, a BIBLIOGRAFIA ACONSELHADA, para se orientar nesta tarefa.

Na primeira parte, FAMILIARIZANDO-SE COM O DISCO , são fornecidas as informações básicas e indispensáveis para que o usuário possa operar o acionador, manipulando os discos de maneira adequada. Nessa parte são ensinados os comandos mais fundamentais, tanto do MSXDOS quanto do DISK BASIC MSX, permitindo operar satisfatoriamente com o equipamento a nível de usuário.

Para os que querem se aprofundar na operação do acionador, aconselhamos a leitura e estudo da segunda parte, PROGRAMANDO COM O DISCO, onde são ensinados os conceitos básicos de programação com os recursos do disco.

Na terceira parte temos dois dicionários extremamente úteis para consultas rápidas, onde os comandos do MSXDOS e do DISK BASIC MSX estão listados em ordem

alfabética. Junto a cada comando encontram-se explicações sobre sua função, sua sintaxe e usos mais comuns.

Para os leitores que não estão familiarizados com os termos MSXDOS e DISK BASIC, cabe uma rápida explicação.

Os computadores MSX já vem de fábrica com a linguagem BASIC gravada de forma permanente em sua memória. Nessa linguagem existem comandos que permitem gravação e leitura de informações em fitas cassete. Além deles, existem comandos que possibilitam o acesso e o controle de acionadores de discos. Estes, entretanto, estão incompletos e "dormentes" e não podem ser usados enquanto uma interface para ligação do acionador não for conectada ao micro. Ao ligarmos o acionador, um programa gravado de maneira permanente em sua interface "desperta" esses comandos completando-os e o usuário passa a dispor de um BASIC mais completo denominado DISK BASIC MSX. Além de poder continuar usando todos os comandos do BASIC original (alguns com funções ampliadas e sintaxe alterada), ele passa a dispor de todos os recursos para operações com discos.

Paralelamente a isto, ao comprar o equipamento, o usuário recebe um disco já gravado, denominado DISCO MESTRE ou DISCO DO SISTEMA que contém um sistema operacional denominado MSXDOS (MSX Disk Operation System). O MSXDOS, ao ser carregado no micro, o transfigura completamente. Quando o MSXDOS está no comando do micro sua organização interna muda de maneira drástica e seus recursos passam a ser completamente diferentes. Esse sistema é compatível, a nível de comandos, com o CP/M e permite a utilização de uma vasta biblioteca de programas desenvolvidos antes da existência do padrão MSX. Como exemplo destes programas poderíamos citar o WORDSTAR, o dBASE, o SUPERCALC, etc.

Consequentemente o leitor precisa ter em mente que, ao ser conectado a um acionador de disco, seu MSX passa a ter uma "dupla personalidade", podendo passar facilmente de uma para a outra mas sem nunca assumir as duas simultaneamente.

CONHECENDO O DISCO

COMO ARMAZENAR INFORMAÇÕES

O seu microcomputador MSX pode armazenar dados e programas de diferentes maneiras.

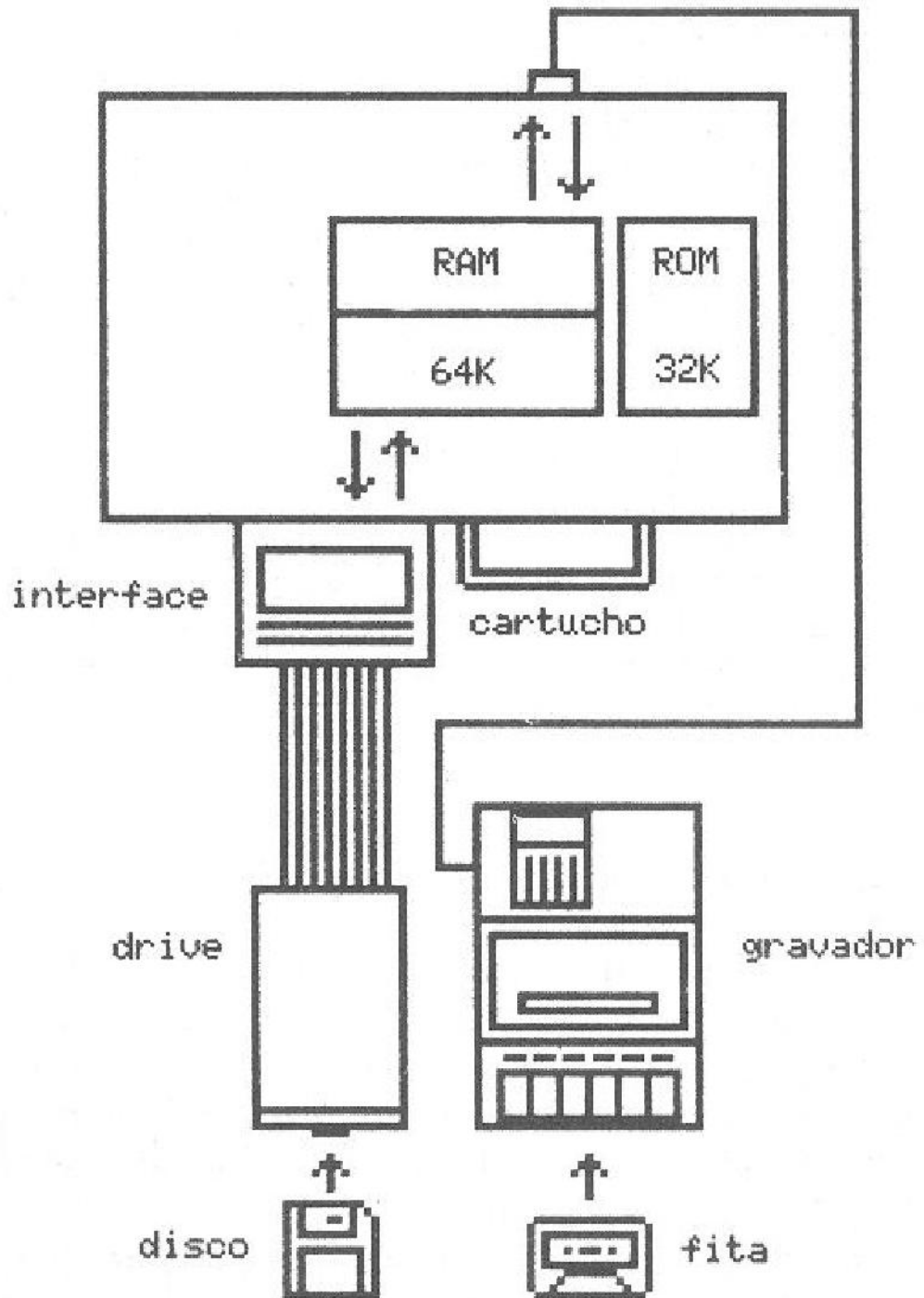
Estes dados (ou programas) podem estar gravados em um circuito eletrônico de forma permanente (ROM). É o que acontece com o Interpretador BASIC e o BIOS, que já vem gravados de fábrica e fazem parte integrante do computador. Esse tipo de gravação é usado também na confecção dos cartuchos que você conecta nos "slots" de seu micro.

No seu computador existe um outro tipo de dispositivo eletrônico denominado memória RAM, no qual você pode gravar e desgravar informações.

Quando você desliga o computador, essas informações são perdidas, pois a memória RAM é completamente desgravada. Para não perder o conteúdo da RAM existem dispositivos externos nos quais você pode "salvar" essas informações. Os mais usados são a fita e o disco magnético. As informações armazenadas podem ser posteriormente recuperadas ao ligarmos o micro, fazendo-o "ler" a fita ou o disco.

Para gravar ou ler informações de uma fita magnética precisamos de um gravador, podendo-se utilizar um equipamento comum de áudio (com algumas restrições). Já para a gravação e leitura de um disco magnético é indis-

pensável um dispositivo especial denominado Acionador de Disco (ou Disk Drive).

FIGURA 1.1 — As "memórias" do computador.

Para que o computador possa "conversar" com o disk drive é necessária uma interface que, no MSX, tem o aspecto de um cartucho e é conectada num dos "slots" da máquina. Essa interface contém circuitos eletrônicos e informações gravadas permanentemente em ROM que lhe permitem controlar o disk drive sob comando do micro.

Lembre-se de NUNCA conectar ou desconectar essa interface (ou qualquer outro tipo de cartucho) com o COMPUTADOR LIGADO!

Apesar do armazenamento em fita ser muito econômico do ponto de vista financeiro, torna-se extremamente dispendioso quando se observa a confiabilidade e o tempo de operação. A gravação e a leitura são muito mais lentos que no disco e o acesso é sequencial. Isso significa que os lotes de informações são gravados um após o outro e a recuperação de alguns deles nos obriga a girar a fita e posicioná-la às cegas, com o auxílio precário do contador de voltas do gravador.

Já no disco, as informações são gravadas e lidas muito mais rapidamente, sendo que a cabeça de leitura e gravação se posiciona automaticamente no local desejado, sem que haja necessidade de se ler todas as informações até se achar a que se busca.

OS DISCOS DO MSX

Existem 2 tipos de discos que podem ser usados com o microcomputador MSX: os de 5 1/4" e os de 3 1/2". Cada um destes tipos possui um acionador característico.

Os discos de 5 1/4" podem ser de face simples (180 KB) ou face dupla (360 KB). Os de 3 1/2" podem ser face simples (360 KB) ou face dupla (720 KB).

Ao contrário do que acontecia antigamente com os discos de Apple, eles não podem ser virados e desvirados feito "panquecas" mas devem ser sempre introduzidos no acionador com a face da etiqueta voltada para cima.

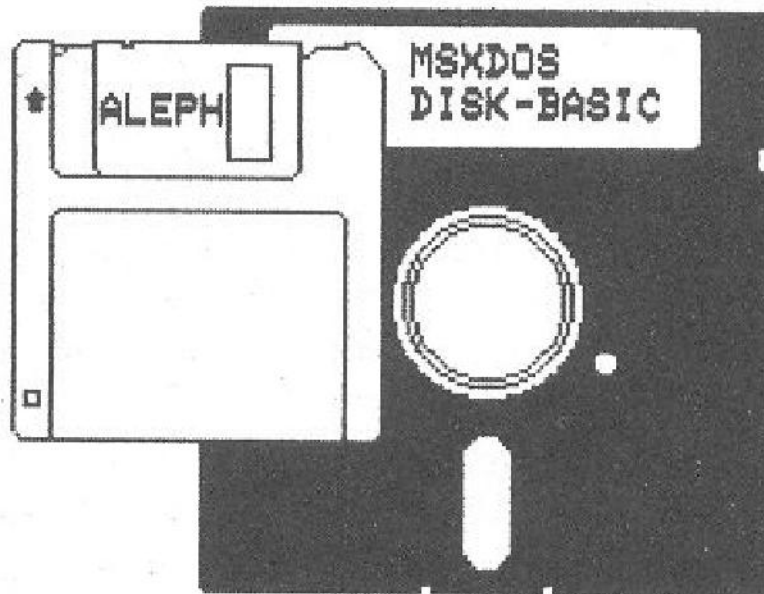
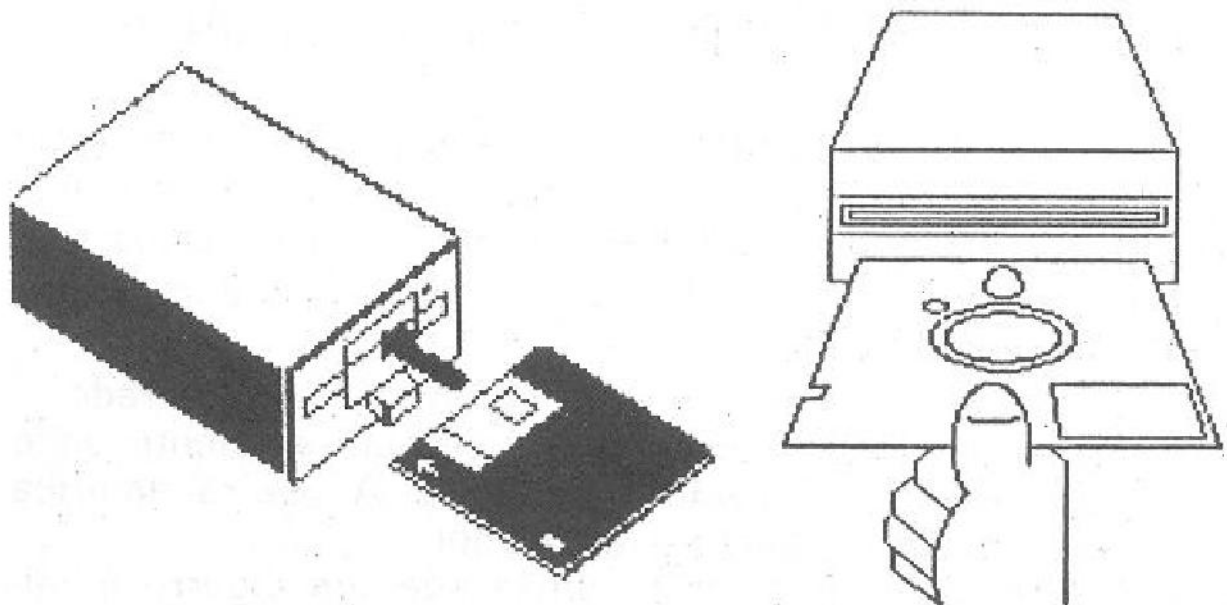
FIGURA 1.2. — Os discos vistos de cima.

Este disco deve sempre ser introduzido no acionador com a face superior voltada para cima. Os "viciados" em discos de 5 1/4", acostumados a virar e desvirar o disco feito panqueca, devem se acostumar a obedecer esta regra básica.

FIGURA 1.3. — Inserção correta dos discos no acionador.

Tanto nos discos de 5 1/4" quanto nos de 3 1/2", ambas as faces estão recobertas de material magnético (que permite a gravação dos dados).

A grande diferença está no fato de que nos discos de face simples (SS = Single Sided) apenas uma das faces está garantida enquanto que nos de face dupla (DS = Double Sided) ambas as faces estão garantidas para gravação de dados.

Da mesma forma, temos dois tipos de acionadores: os que têm apenas uma cabeça de gravação/leitura, nos quais podemos usar discos SS e os que têm duas cabeças, nos quais devemos usar discos DS.

Note que, numa emergência, podemos usar discos SS num acionador com duas cabeças: nem sempre a não garantia numa das faces implica na existência obrigatória de defeitos.

Além disso, podemos usar um acionador de duas cabeças para gravar informações para um amigo que tenha um acionador de face simples. Como veremos mais tarde, basta informar ao acionador que queremos discos de face simples e ele operará com apenas uma cabeça, simulando um drive de face simples.

CUIDADOS COM O DISCO E COM O DRIVE

Junto com seu acionador, você deve ter recebido uma interface controladora e um disco já gravado (disco-mestre). Antes de manipulá-lo, lembre-se que seus discos têm vários inimigos, alguns óbvios e outros invisíveis. Entre eles podemos relacionar:

1 — POEIRA — Não trabalhe em ambiente empoeirado. Procure instalar seu computador num ambiente onde o ar esteja razoavelmente limpo. A poeira danifica tanto o disco quanto o acionador.

2 — FUMAÇA — A própria fumaça de um cigarro é ini-

migo mortal dos discos magnéticos. Não fume perto do computador. Isso fará muito mal aos discos e mais ainda à sua saúde!

3 — CALOR — Não exponha os discos ao calor e ao Sol!

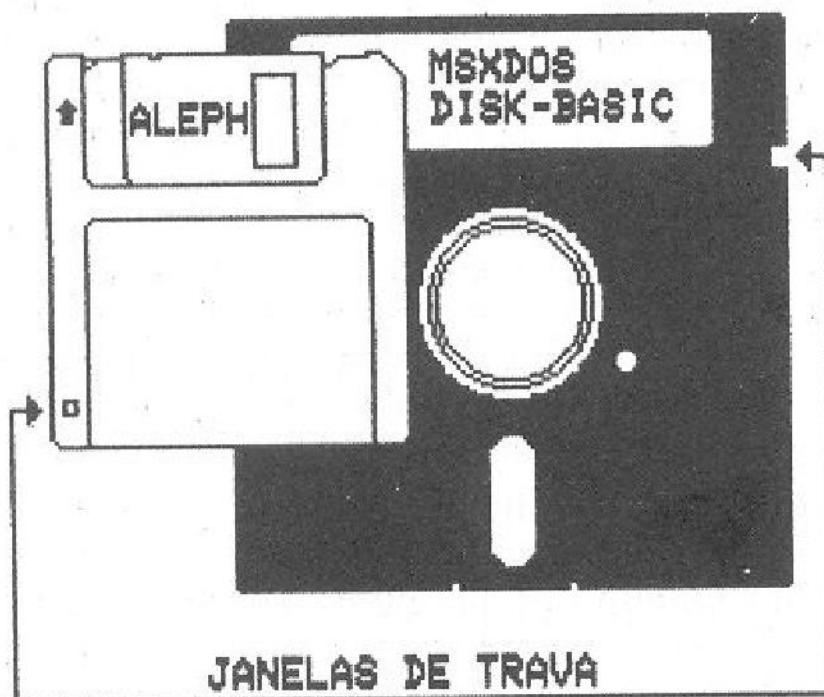
4 — UMIDADE — Evite ambientes úmidos e jamais molhe os discos (ou espirre sobre eles).

5 — PULSOS NA REDE ELÉTRICA — Mesmo que a rede elétrica à qual seu micro e seu acionador estão conectados não sofra variações grandes de tensão, existe um risco de dano grave. Se a tomada estiver em paralelo com um circuito de grande indutância (como por exemplo, reatores de lâmpadas fluorescentes), ao desligarmos ou ligarmos outros equipamentos da rede produzimos picos de tensão que podem danificar o micro e o acionador. Para evitar esse inconveniente é bom que o micro e o acionador estejam ligados a um transformador regulador de voltagem automático, para filtrar esses picos. Dessa forma evitaremos que uma simples partida numa geladeira ou o acendedor de lâmpadas fluorescentes provoque danos no equipamento.

6 — CAMPOS MAGNÉTICOS — Como a gravação das informações no disco se dá por processo magnético, qualquer campo intenso pode danificar o conteúdo do disco. Os campos magnéticos são gerados por imãs, que podem ser facilmente identificáveis (como os clássicos em forma de ferradura) ou disfarçados (na ponta de uma tesoura, num prendedor de clips, ou num fecho de armários). Além disso, campos magnéticos são gerados quando uma corrente passa por um fio condutor, especialmente se ele estiver enrolado na forma de bobina. Esses enrolamentos podem ser encontrados em transformadores, motores elétricos, reatores de lâmpadas fluorescentes ou de neon, etc. Consequentemente, tome cuidado não só com a localização do equipamento, mas também com o local de armazenamento ou transporte dos discos. As vezes uma viagem de ônibus elétrico pode tornar seus discos ilegíveis!

PRIMEIRAS OPERAÇÕES COM DISCO

Antes de começarmos a operar com o disco, existe uma precaução básica extremamente útil: seu disco pode ser "travado" de maneira a não aceitar gravação de dados adicionais. Dessa forma, mesmo que você cometa algum erro na digitação de algum comando, isso realmente não resultará em danos no disco.

FIGURA 1.4 — A "trava" contra gravação.

Nos discos de 5 1/4" existe uma janela na parte lateral que deve ser vedada com uma pequena etiqueta auto-adesiva para que o disco fique travado (permitindo apenas a leitura de dados mas não a gravação).

Nos de 3 1/2", há, na parte posterior do disco, uma janela com uma tampinha de plástico que pode ser deslizada (com a ajuda de um clip) de maneira a abri-la ou fechá-la. Estando aberta, o disco está travado.

Cuidado: a filosofia é oposta nos dois tipos de discos:

5 1/4"	aberta	destravado
3 1/2"	aberta	travado

Vamos finalmente começar as operações com disco.

1 — Certifique-se que o micro e o acionador estejam desligados e conecte a interface num slot do MSX, preferencialmente o "A".

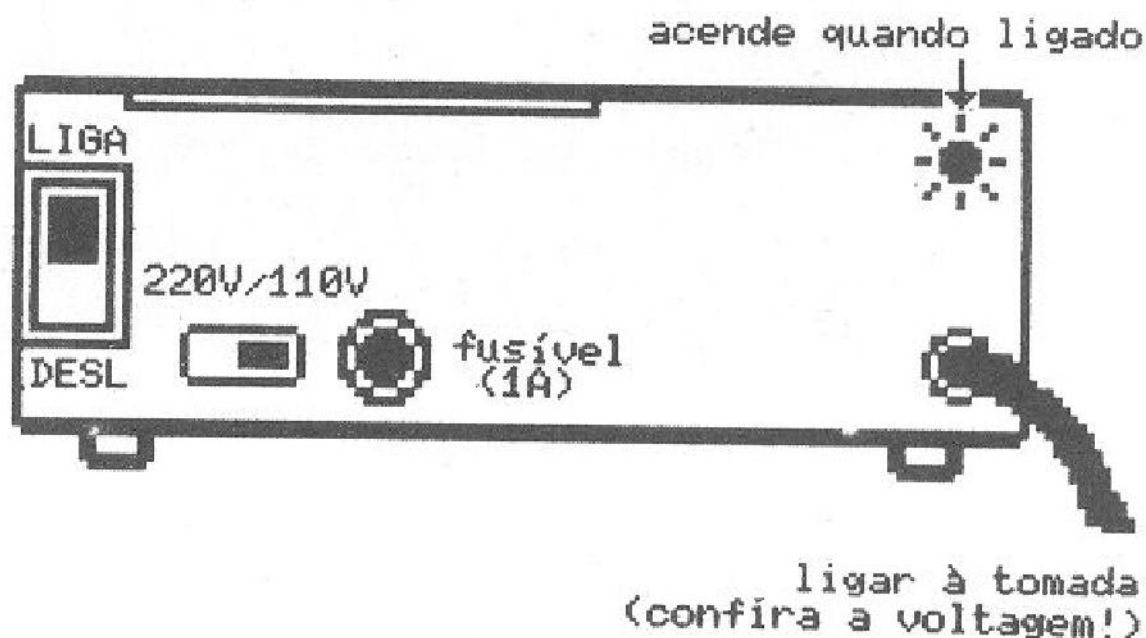
2 — Conecte todos os periféricos adicionais que você for usar (impressora, modem, etc). Se você for usar um cartão de 80 colunas, não o ligue por enquanto para poder acompanhar melhor a explicação a seguir.

3 — Coloque o disco do sistema (o disco já gravado que vem junto com o equipamento), TRAVADO (janela ABERTA) na frente do acionador com a face superior voltada para cima (reveja a figura 1.3) e empurre-o delicadamente para dentro até sentir que está retido. Dê mais um empurrãozinho, abaixando um pouco o disco até fazer saltar o botão de ejeção para fora. Lembre-se, a setinha gravada na caixa plástica do disco deve estar do lado esquerdo, apontando para dentro da fenda.

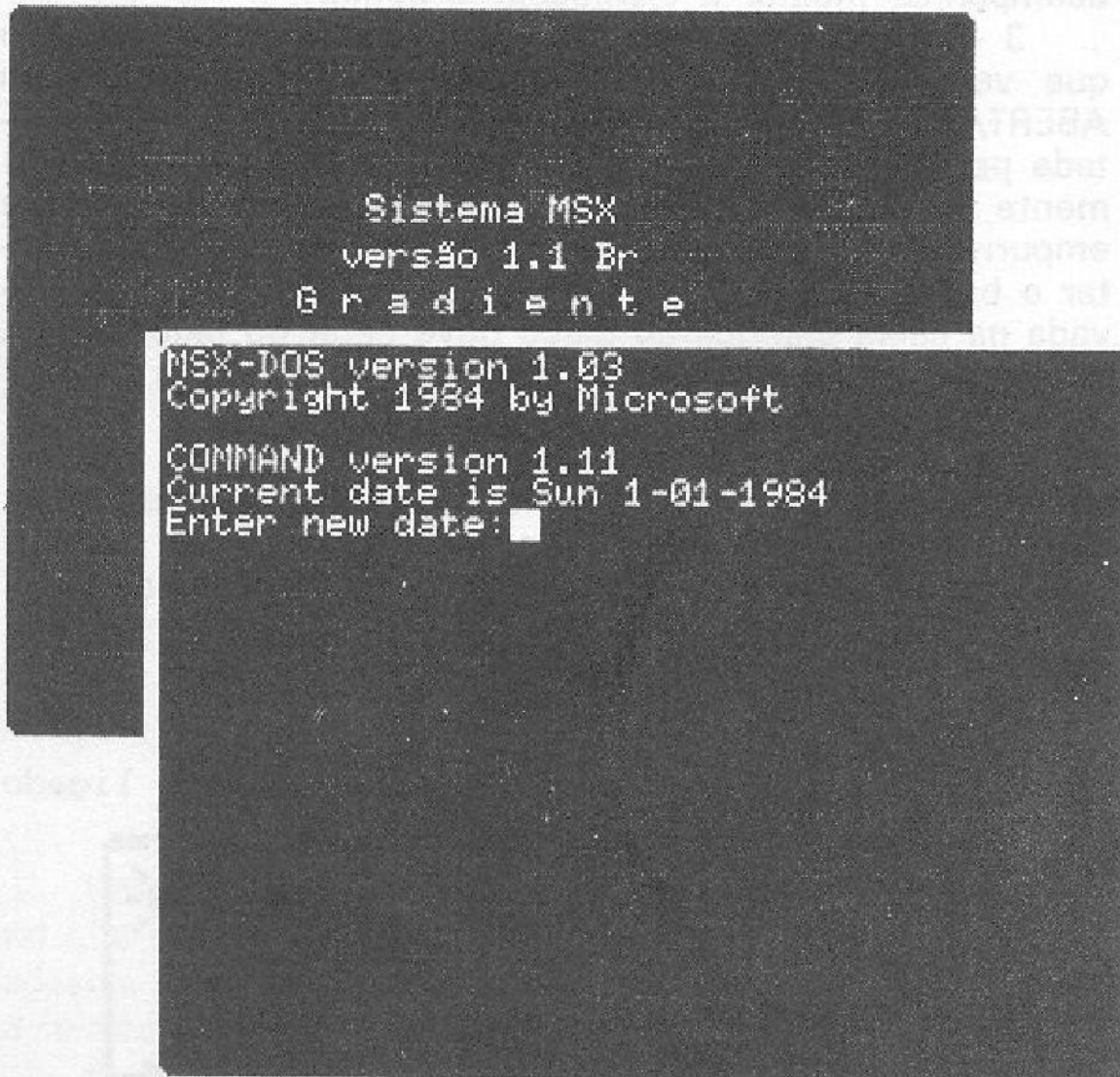
4 — Ligue o vídeo.

5 — Ligue outros periféricos.

6 — Ligue o acionador no interruptor localizado na parte traseira (figura 1.5). Você verificará que a luz vermelha localizada na parte traseira do acionador acenderá.

FIGURA 1.5 — Como ligar o acionador.

7 — Ligue o microcomputador.

Na tela do vídeo deverá aparecer a mensagem do fabricante do MSX e, a seguir a mensagem do MSXDOS assumindo o controle do micro (figura 1.6).

FIGURA 1.6 — As telas de apresentação.

A luz no painel frontal do acionador deverá acender durante um breve intervalo, indicando que o disco está sendo acessado.

Se aparecer a mensagem:

```
Entre data (D-M-A):█
```

significa que o disco não foi bem introduzido e o micro

não conseguiu o MSXDOS a partir dele. Desligue o equipamento e tente outra vez, inserindo o disco corretamente (o botão de ejeção deve saltar para fora).

Se a tela da figura 1.6 não aparecer, desligue tudo na ordem inversa, verifique as conexões de todos os cabos em seus equipamentos, a inserção do disco, e tente ligar o equipamento mais uma vez.

Tendo a tela da figura 1.6 no vídeo de seu micro, aperte RETURN (não se preocupe em atualizar a data por enquanto).

Ao fazer isso, você deverá ver o sinal de PRONTO do MSXDOS seguido do cursor, mostrando que o sistema está esperando a digitação de algum comando:

```
A>█
```

Vamos aprender nosso primeiro comando do MSXDOS: o MODE. Ele é parecido com o WIDTH do BASIC e permite determinar o número de colunas que a tela deve apresentar.

Como em alguns televisores a largura de 40 colunas é excessiva (não permitindo a visão das colunas laterais), digite:

```
A>mode 39
```

1 — Não se preocupe com maiúsculas, pois o MSXDOS também reconhece minúsculas.
2 — Não esqueça de deixar um espaço entre o comando MODE e o parâmetro 39. Se você errou algo é só usar a seta para a esquerda (que no MSXDOS faz o papel de BACK SPACE, ou seja, volta apagando).
3 — Após digitar o comando, para que ele seja executado, pressione a tecla RETURN.

A tela deverá ser apagada e o sinal de PRONTO com o cursor deve se posicionar no canto superior direito.

Se você quiser brincar um pouco, digite:

```
A>MODE 20
```

Veja como o cursor se posicionou, mas depois, volte ao MODE 39. Se você é uma pessoa curiosa deve ter ex-

perimentado o MODE 1. Volte agora ao MODE 39, não esquecendo de digitar um espaço entre o comando e o parâmetro.

Vamos agora ver um dos comandos mais importantes: o DIR. Este comando, ao ser utilizado, faz com que o micro leia o diretório do disco, uma espécie de índice de seu conteúdo. Digite então:

```
A>dir
```

Não esqueça de pressionar RETURN após cada comando! O MSXDOS só os executará após você pressionar essa tecla.

A luz vermelha no painel frontal do acionador deverá acender, mostrando que o micro está lendo algo do disco (pressionar o botão de ejeção enquanto essa luz está acesa poderá ter um efeito destruidor no conteúdo do disco!).

A seguir será listado na tela o conteúdo do disco.

FIGURA 1.7 — O diretório do disco mestre.

```
A>DIR
MSXDOS   SYS     2560 20-11-87   9:07a
COMMAND  COM     6656 20-11-87   7:56a
FUNKEY   BAT       19 10-11-87
FUNKEY   BAS     1745 10-11-87
ESTATIST BAS     2408 10-11-87
MUSICA   BAS     2678 10-11-87
LISTPRE  BAS     1482 10-11-87
COPYSCR  BAS     2005 10-11-87
COPYSCR  BIN      608 10-11-87
EDITCAR  BAS     4460 10-11-87
EDITCAR  BIN     2807 10-11-87
BOLD     TAB     2056 10-11-87
        12 files   326656 bytes free
A>█
```

Na primeira coluna você tem o nome dos arquivos gravados no disco. Na segunda coluna você tem um código de três letras que indica o tipo de arquivo. Maiores

detalhes sobre estes códigos serão vistos no capítulo 5. Na terceira coluna é fornecido o número de bytes de cada arquivo. Por exemplo, os arquivos

```
MSXDOS   SYS
COMMAND  COM
```

ocupam respectivamente 2432 e 6656 bytes.

Na quarta coluna temos a data em que o arquivo foi gravado e na quinta coluna, o horário. É conveniente observar que apenas microcomputadores dotados de relógio (interno ou em cartucho) tem condições de atualizar o instante de gravação.

Na última linha do diretório é relatado quantos arquivos foram gravados e quantos bytes livres ainda estão disponíveis no disco.

Quando o diretório é muito longo, às vezes torna-se difícil achar um determinado arquivo. Porisso você pode pedir para listar um arquivo específico. Por exemplo, experimente digitar:

```
A>dir msxdos.sys                (e RETURN !)
```

Você verá que apenas um arquivo será listado na tela:

```
MSXDOS   SYS     2560 20-11-87  9:07a
```

Se você não lembrar o nome do arquivo que quer listar, mas souber que ele é seguido do código BAT, por exemplo, bastará digitar:

```
A>dir *.bat                     (e RETURN !)
```

Você obterá a listagem de todos os arquivos que têm o código BAT em seu final. O asterisco fez o papel de "coringa": é como se você tivesse dito ao computador "me liste o diretório de todos os arquivos cujo nome seja "alguma coisa" seguida de BAT.

COMO TROCAR INFORMAÇÕES ENTRE DISCOS

BACKUP DE DISCO

Vamos agora aprender uma coisa importantíssima: o BACKUP de um disco, ou seja, sua cópia fiel de maneira a não perder informações se algum acidente acontecer com o disco original.

Você já deve ter ouvido falar na LEI DE MURPHY: "Se alguma coisa tem a mais remota chance de dar errado, certamente dará errado!"

Apesar dela ter sido concebida com intuito humorístico, sua validade no mundo da informática é confirmada por milhões de usuários levados às raias do desespero por faltas de energia elétrica, sobrinhos curiosos e distrações fatais.

Por isso o BACKUP de discos que contenham informações importantes é fundamental.

Inicialmente vamos partir do pressuposto que você tenha uma única interface de disco conectada ao micro e que ela comande um único acionador.

Coloque o disco mestre no acionador certificando-se de que ele esteja com a janelinha ABERTA. Ligue o sistema como descrito, de maneira a obter o:

Neste momento o micro está sob o controle do MSXDOS. Se necessário, comande:

```
A>mode 39            (e RETURN!)
```

para visualizar a primeira coluna.

Pegue agora um disco virgem. Vamos prepará-lo para que ele possa receber informações do seu MSX.

Retire o disco mestre do acionador (pressionando o botão de ejeção) e destrave o disco virgem FECHANDO a janela de trava. Coloque-o no acionador com a face superior sempre voltada para cima e empurre-o o suficiente para fazer o botão de ejeção saltar para fora.

Digite agora:

```
A>FORMAT
```

Aparecerá a pergunta:

```
Nome do drive? (A,B) █
```

Como estamos supondo que apenas um drive está conectado ao sistema, digite:

```
A
```

Aparecerá um "menu" de quatro opções (figura 2.1).

FIGURA 2.1 — Menu de formatação.

```
Nº Faces  Trilhas
1→  1      40
2→  2      40
3→  1      80
4→  2      80
```

Estas opções se referem às seguintes situações:

1 — 5 1/4" face simples
2 — 5 1/4" face dupla
3 — 3 1/2" face simples
4 — 3 1/2" face dupla

Se você tem um acionador de face dupla mas quer preparar um disco para alguém que tem um drive de face simples, use a opção do seu drive em face simples. Note que existem acionadores de 5 1/4" importados que permitem a gravação de 80 trilhas. Neste caso use as opções de 3 1/2"!

```
Digite algo quando pronto █
```

Verifique mais uma vez se o disco está destravado e bem colocado no acionador (lembre-se da LEI DE MURPHY).

Pressione a barra de espaços e você verá a luz frontal do acionador se acender, indicando que o computador está escrevendo algo no disco.

Na realidade o disco está recebendo informações de uso interno, de maneira a poder ser reconhecido como um disco MSX pelo seu acionador. Este processo é chamado FORMATAÇÃO e será visto com mais detalhes no capítulo 3.

Quando a luz se apagar, aparecerá a mensagem:

```
Formatação completa
```

Você obterá outra vez o sinal de "pronto" do MSXDOS.

Seu disco agora está "vazio" (ou seja, não contém informações gravadas pelo usuário) mas não está mais virgem, ou seja, já contém todos os dados de uso interno do micro e da interface de maneira a ser reconhecido, daqui para frente, como um disco de 3 1/2" ou 5 1/4", face simples ou dupla, dependendo da opção que você escolheu no menu de formatação.

Retire-o do acionador (apertando o botão de ejeção) e coloque o disco mestre, certificando-se de que ele esteja travado (janela ABERTA).

Para verificar mais uma vez quais arquivos ele contém, comande:

```
A>DIR                    (e RETURN!)
```

Se ele não estiver bem colocado, aparecerá a mensagem:

```
Not ready error reading drive A
Abort, Retry, Ignore? █
```

Neste caso recoloque-o corretamente no acionador e tecle R (de Retry).

Após a listagem do diretório, você obtém outra vez o "pronto".

Digite, com muito cuidado e respeitando os espaços, o seguinte comando:

```
A>COPY A:*.* B:/V
```

Antes de pressionar a tecla RETURN, vamos entender

o tipo de ordem dada ao computador. O que acabamos de comandar pode ser interpretado como:

A>COPY A:	copie do drive A
.	os arquivos que tenham qualquer nome e qualquer código de 3 letras (ou seja: TODOS)
B:	para o drive B
/V	e verifique se a cópia foi perfeita.

Neste momento você poderá objetar "espera aí: eu só tenho um drive ligado na interface!"

Não se preocupe: a interface usará o único acionador conectado como drive A e drive B alternadamente durante o processo de cópia. Basta ficar atento às mensagens na tela que pedem a substituição do disco no acionador.

Agora digite RETURN. Na tela você verá os arquivos que foram lidos do disco e foram carregados na memória do micro. A seguir você lerá a mensagem:

```
Inserir disquete para drive B:
E digite algo quando pronto█
```

Isso significa que a interface deixou de considerar o acionador conectado como drive A e passou a tratá-lo como drive B.

Retire o disco mestre do acionador (que agora faz papel de drive B) e coloque o disco formatado (e DESTRAVADO).

Se todos os arquivos pedidos forem copiados, aparecerá uma mensagem dizendo:

```
12 files copied
```

Novamente, o sinal de pronto (A>) será apresentado.

Se ainda há outros arquivos no disco original a serem copiados, o computador continuará solicitando a troca de discos (cuidado para não fazer confusão) até que o processo se complete.

Agora você tem um BACKUP do seu disco mestre. Guarde o disco mestre em algum lugar seguro, travado, e não mexa mais nele a não ser que aconteça algum aci-

dente com o disco mestre que você acabou de gerar.

Trave o disco mestre de uso (janela ABERTA) e use-o apenas para leitura do MSXDOS. Para usar como disco de trabalho, e acompanhar o resto deste livro, seria bom formatar outro disco virgem e repetir o processo de BACKUP, deixando-o destravado de maneira a poder gravar informações suplementares.

É importante etiquetar convenientemente todos os discos novos de modo a poder identificá-los rapidamente.

Neste momento você já deve ter ao menos 3 discos formatados:

DISCO MESTRE ORIGINAL (TRAVADO E GUARDADO)
DISCO MESTRE COPIADO (TRAVADO E EM USO)
DISCO DE TRABALHO (DESTRAVADO E EM USO)

Aprenda a se organizar e muito tempo e dissabores serão poupados!

FACE SIMPLES E FACE DUPLA

É perfeitamente possível se fazer uma cópia de arquivos de um disco de 5 1/4" para outro de 3 1/2" (ou vice-versa), ou entre dois discos de 3 1/2", sendo um face simples e outro face dupla.

Porém é indispensável termos sempre presente algumas informações básicas.

Um acionador (drive) de face simples só pode ler e gravar informações em discos formatados como face simples (SS).

Em contrapartida, um acionador de face dupla pode funcionar das duas maneiras: tanto formata discos em face dupla, quanto em simples, dependendo da opção escolhida no menu de formatação.

Quando colocamos nele um disco formatado como face simples, ele o reconhece como tal (esta, aliás, é uma das razões da formatação) e passa a ler e gravar nele como se fosse um acionador de face simples (usando apenas a cabeça superior).

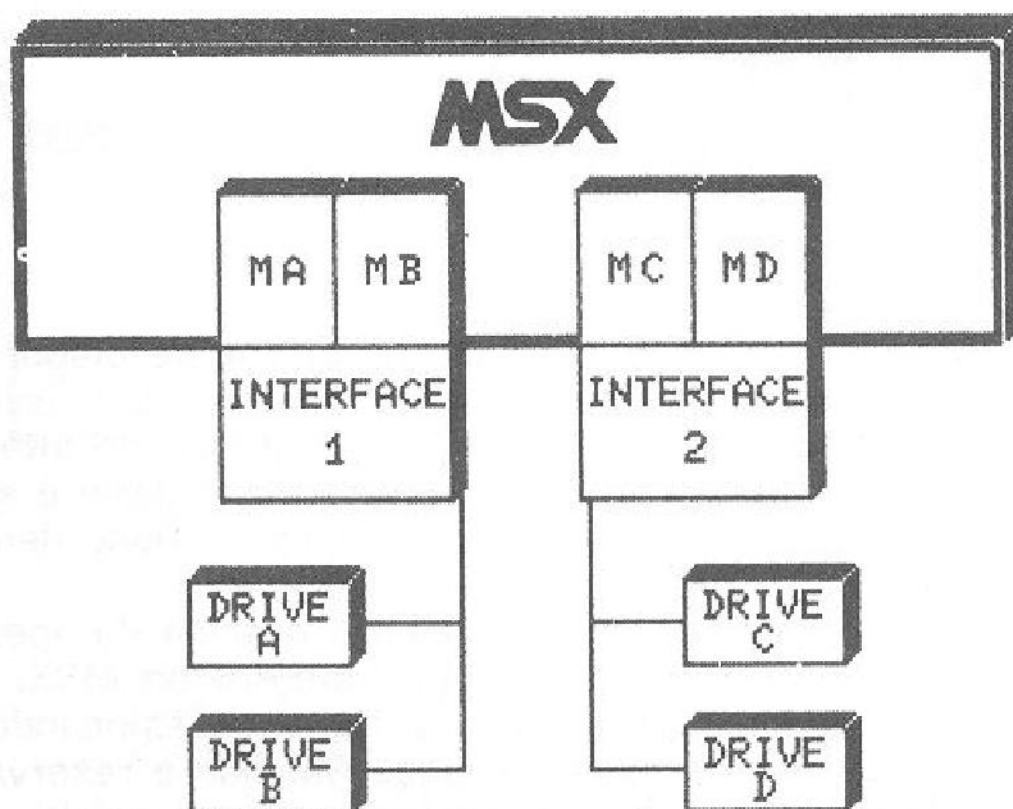
OPERANDO COM VÁRIOS ACIONADORES E INTERFACES

A cada interface padrão MSX podem ser conectados dois acionadores de disco (iguais ou diferentes) de 5 1/4" ou 3 1/2".

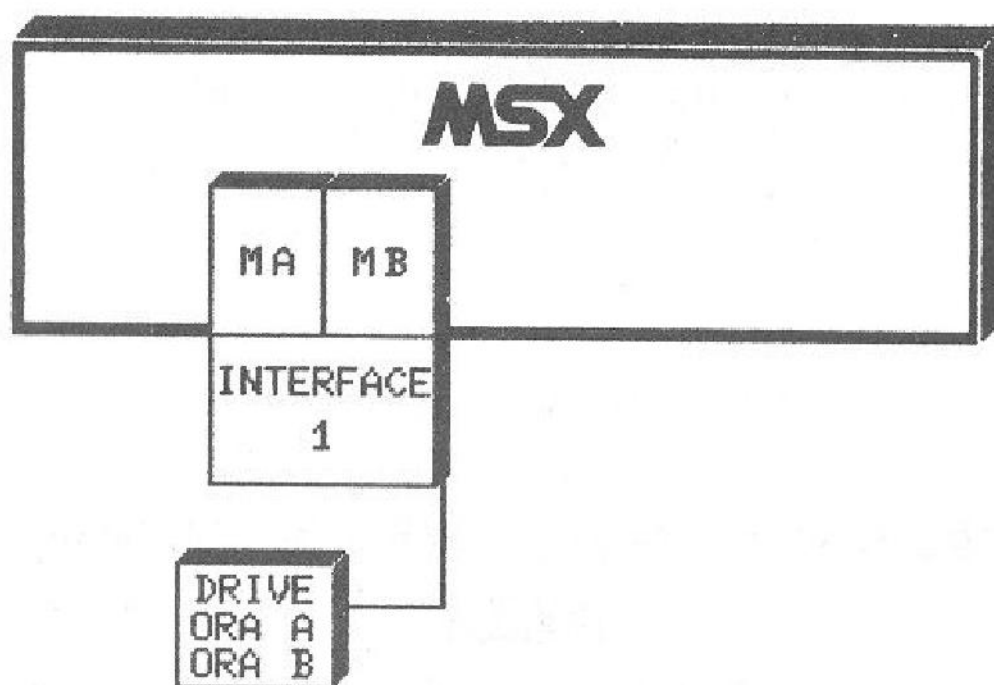
Quando temos uma única interface ligada a dois drives, ela trata um deles como A e outro como B.

Como podemos conectar 2 interfaces padrão MSX no micro (uma em cada slot), podemos operar simultaneamente 4 acionadores, sendo o terceiro e o quarto denominados C e D.

Para o uso de operação de cada acionador a interface reserva um trecho da memória do MSX. Vamos chamar cada uma dessas porções de memória de MA, MB, MC e MD.

FIGURA 2.2 — O MSX ligado a quatro acionadores.

Note que, se os 4 acionadores forem 80 trilhas-face dupla, você poderá gerenciar, simultaneamente nada mais nada menos que 2,8 megabytes de memória em disco!

Se você tiver uma única interface ligada a um único acionador, assim mesmo são reservadas as porções de memória MA e MB no MSX.

FIGURA 2.3 — O MSX ligado a uma interface e um acionador.

Isso ocorre para que a interface, apesar de dispor de apenas um drive físico, possa operá-lo como dois drives lógicos. Obviamente, quando o acionador físico se alterna nos papéis ora de drive A, ora de drive B, o usuário é solicitado, sempre que for necessário, a trocar o disco dentro do acionador.

Em alguns casos, porém, o usuário precisa de apenas um drive lógico e mais espaço na memória do MSX. Ao ligar o computador, basta que ele fique pressionando a tecla CONTROL. Neste caso o espaço MB não é reservado na memória do MSX e fica disponível para o usuário.

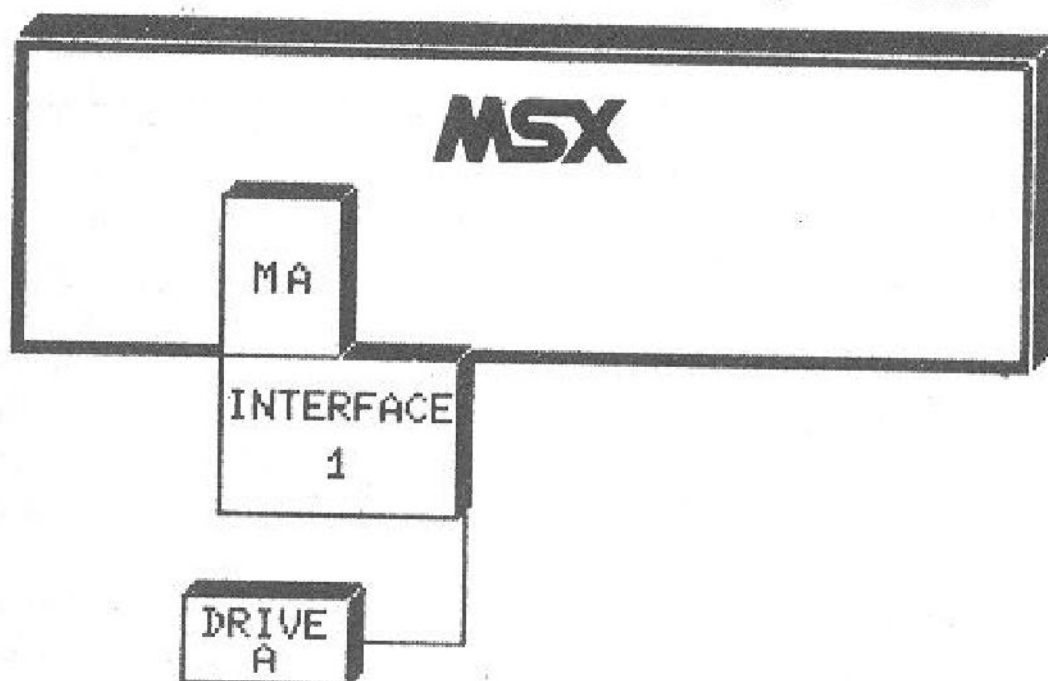
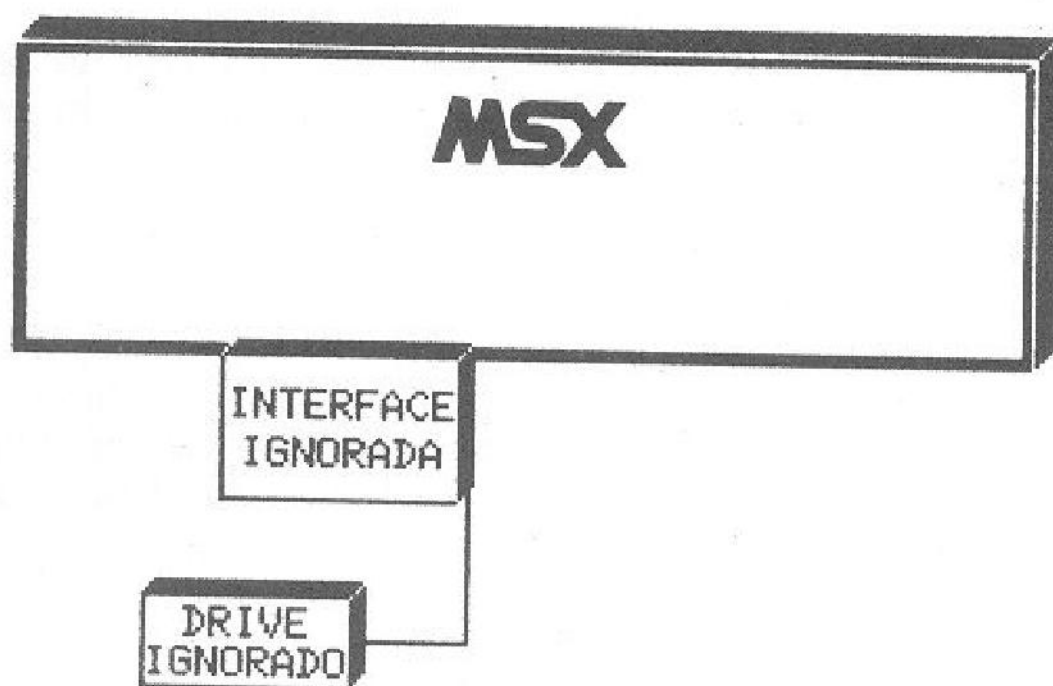
FIGURA 2.4 — Ligando o Micro com CONTROL pressionada.

Em certos casos, deseja-se utilizar o micro sem a interface de disco ligada. Porém, ficar ligando e desligando cartuchos do slot com muita frequência provoca um desgaste prematuro nos contatos. Basta então, ao ligarmos o computador, ficar pressionando a tecla SHIFT.

FIGURA 2.5. — Ligando o Micro com a tecla SHIFT pressionada.

Neste caso o MSX ignora a presença da interface sem que haja necessidade de desconectá-la do sistema.

AS "TRANSFUSÕES" ENTRE 5 1/4" E 3 1/2"

Quando queremos copiar arquivos de um disco para outro do mesmo tipo, já vimos que não há problemas. Aliás, já fizemos isto no início deste capítulo ao gerarmos o disco mestre de uso e o disco de trabalho (BACKUP).

Porém, se quisermos passar um arquivo de um disco de 5 1/4" para outro de 3 1/2" (ou vice-versa), é indispensável que os dois acionadores estejam conectados simultaneamente ao micro.

Como ligar dois acionadores a uma única interface dá um certo trabalho (precisamos abrir a caixa do acionador para configurá-lo) o mais aconselhável é se ter uma interface para operar o drive de 3 1/2" e outra, ligada ao outro slot, para comandar o drive de 5 1/4".

Assim fazendo um dos drives (por exemplo o de 3 1/2") vai operar como A (ou B) e o outro (no caso, de 5 1/4") funcionará como C (ou D).

Com o DOS carregado podemos, por exemplo, comandar:

```
A>COPY A:BABA.GUT C:CACA.NIC
```

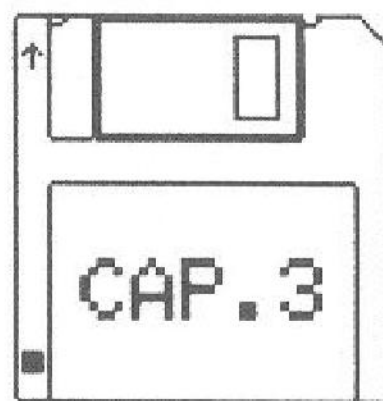
Assim fazendo o arquivo BABA.GUT será lido no drive A (de 3 1/2") e gravado com o nome CACA.NIC no drive C (de 5 1/4").

Note que, se você comandar:

```
A>COPY A:BABA.GUT B:CACA.NIC
```

o sistema solicitará uma troca de discos no drive de 3 1/2" pois ele está sendo usado como dois drives lógicos.

Lembre-se que, se o sistema for inicializado com a tecla CONTROL pressionada, cada interface controlará apenas 1 drive!

INTRODUÇÃO AO MSXDOS

A PREPARAÇÃO DO DISCO PARA O USO

Vimos no capítulo anterior que para poder gravar qualquer coisa num disco é necessário "formatá-lo". Vamos agora aprender um pouco mais sobre o modo como os dados são armazenados nos discos e em que, exatamente, consiste a formatação.

Imagine um disco como um caderno. Assim que ele é adquirido ainda está "virgem", como um caderno que acabou de ser comprado numa loja, sem nada escrito.

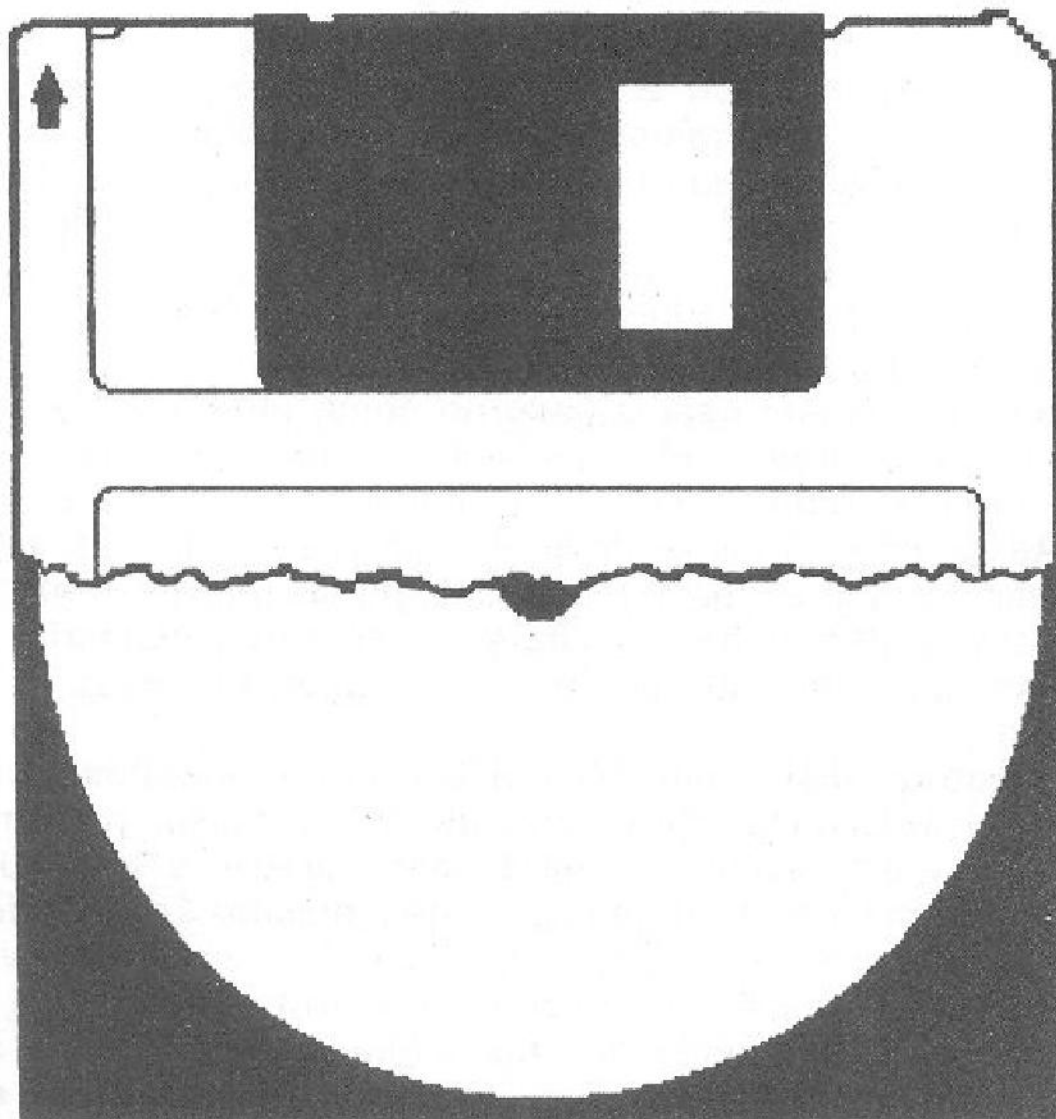
Se você quiser usar o caderno como uma agenda será necessário numerar todas as suas páginas e ainda colocar a data no topo de cada uma delas.

As páginas iniciais deverão ser reservadas para armazenar um índice da agenda e algumas outras informações que poderão lhe ser úteis, como um calendário do ano anterior e do ano seguinte, seu nome, endereço, CIC, RG, etc.

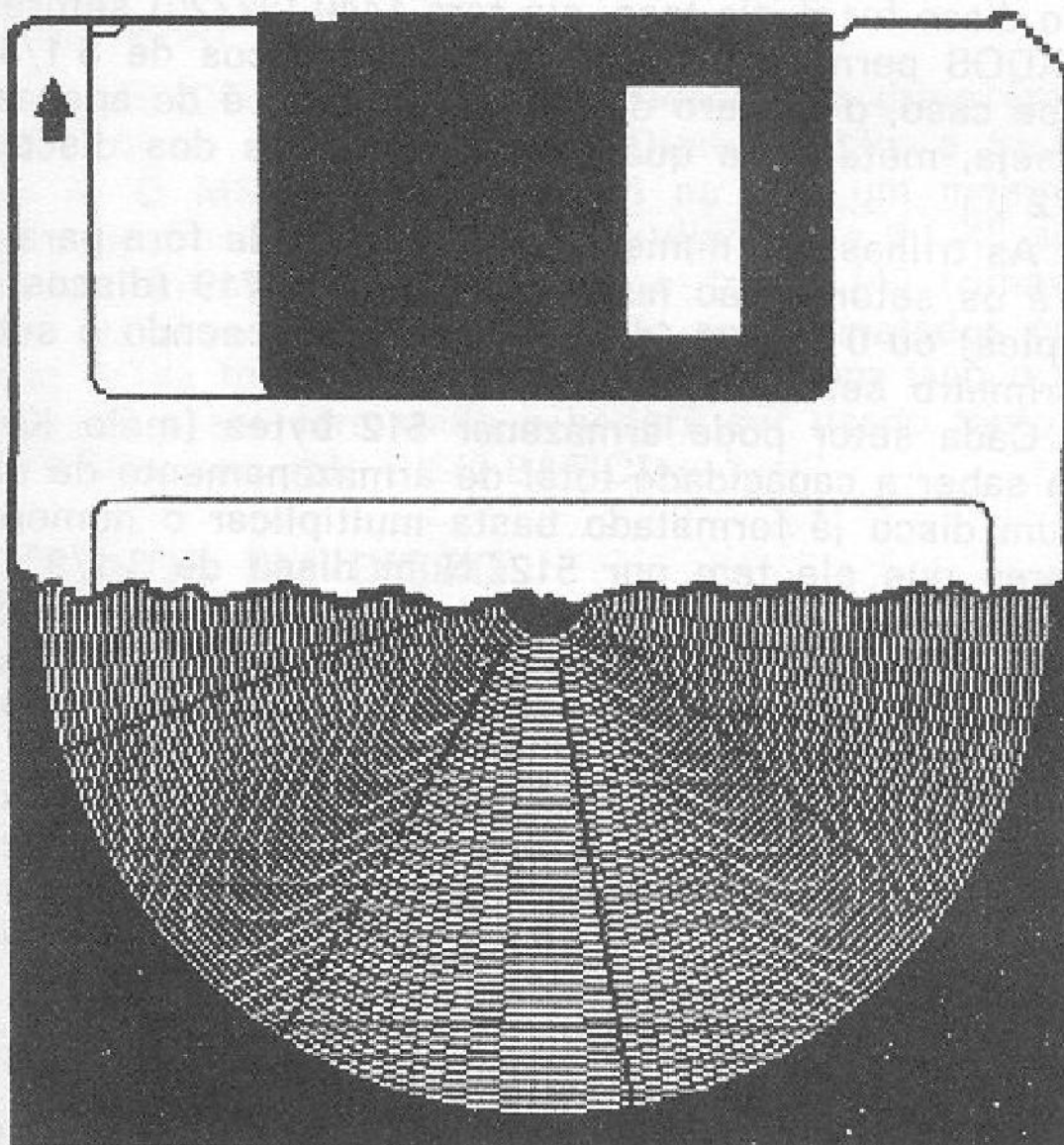
O uso do disco pelo MSXDOS é muito semelhante ao uso de um caderno como agenda. Ele precisa, portanto, de informações adicionais para poder gravar e ler dados de uma forma rápida e eficiente nos discos. Essas informações tem que ser escritas no disco ao menos uma vez antes que ele possa ser usado para armazenar dados e correspondem à numeração das páginas, ao índice e às demais informações úteis que descrevemos para o uso de um caderno como agenda.

O disco, apesar das analogias que podemos fazer, é bem diferente de um caderno e a "formatação" é muito mais complexa que uma simples numeração de páginas, gastando uma quantidade apreciável de capacidade de armazenamento do disco. É por isso, por exemplo, que num disco face simples (que armazena 500 Kbytes), o usuário pode gravar (depois de formatado) 360 Kbytes de informações.

Um disco não formatado é apenas uma lâmina de plástico circular revestida em ambas as faces com material magnetizável, de modo semelhante à uma fita cassete.

FIGURA 3.1 — Disco não formatado.

Ao ser formatado, apenas parte da superfície do disco passa a ser realmente útil para armazenar dados. Ele é gravado em várias circunferências concêntricas chamadas "trilhas". Apenas as trilhas são usadas. As regiões entre elas, quando existem, nunca são gravadas ou lidas.

Dentro de cada trilha, é gravada uma grande quantidade de informações de controle e a trilha é ainda dividida em vários pedaços, chamados "setores".

FIGURA 3.2 — Disco formatado.

Para cada setor, mais dados de controle são gravados no disco. Desse modo, após ser formatado, ele está repleto de informações de controle para que o MSXDOS e o DISK BASIC possam usá-lo de forma bastante eficaz.

Durante a formatação, além de serem gravadas, todas as trilhas do disco são lidas para verificar se há algum problema com ele.

Note que mesmo os discos já usados podem ser formatados! Portanto, antes de formatar um disco, certifique-se que ele esteja virgem ou que os dados que ele contém podem ser perdidos.

O MSXDOS formata os discos de 3 1/2" em 80 trilhas por face, sendo cada trilha dividida em 9 setores. Temos, portanto, um total de 720 setores em cada face do disco. Se o disco for dupla face, ele terá 1440 (2x720) setores! O MSXDOS permite também o uso de discos de 5 1/4" e, nesse caso, o número de trilhas por face é de apenas 40, ou seja, metade da quantidade de trilhas dos discos de 3 1/2".

As trilhas são numeradas de 0 a 80 de fora para dentro e os setores são numerados de 0 a 719 (discos face simples) ou 0 a 1439 (discos face dupla), sendo o setor 0 o primeiro setor da trilha 0.

Cada setor pode armazenar 512 bytes (meio Kbyte). Para saber a capacidade total de armazenamento de dados de um disco já formatado basta multiplicar o número de setores que ele tem por 512. Num disco de 3 1/2" face simples podemos armazenar até 360 Kbytes e num disco de 3 1/2" face dupla podemos armazenar até 720 Kbytes! Os discos de 5 1/4" são formatados de modo análogo aos de 3 1/2", porém com 40 trilhas por face.

Além da divisão do disco em trilhas e setores, a formatação reserva os onze primeiros setores para armazenar um pequeno programa e um índice de todos os dados que forem gravados nele. Os setores são divididos da seguinte forma:

setor 0 — Dados gerais do disco e inicialização.
setor 1 ┐ Tabela contendo a relação dos setores
setor 2 │ usados para cada programa ou conjunto
setor 3 │ de dados gravados no disco.
setor 4 ┘
setor 5 ┐ Nestes setores localiza-se o índice
setor 6 │ geral do disco, chamado DIRETÓRIO.
setor 7 │ Sempre que algum dado é gravado ou lido

setor 8 — no disco, o diretório é atualizado.
setor 9 — No diretório podem ser relacionados até
setor 10 — 112 programas ou conjuntos de dados.
setor 11 —

Os demais setores do disco ficam livres para receber dados.

O comando do MSXDOS que executa a formatação de um disco é o FORMAT. Experimente formatar alguns discos virgens. Basta comandar:

```
A>FORMAT
```

O MSXDOS pedirá a especificação do drive em que está o disco a ser formatado. Digite, então, a tecla da letra A. O MSXDOS apresentará na tela um menu com quatro opções. Lembre-se que os discos de 5 1/4" só podem ser formatados pelas opções 1 e 2 (40 trilhas por face). Os discos de 3 1/2" devem ser formatados com a opção 3 (se for SS) ou 4 (se for DS). Com isso o disco começará a ser formatado e poderá ser usado tanto pelo MSXDOS como pelo DISK BASIC).

ARQUIVOS EM DISCO

Após formatar alguns discos, insira novamente o disco com o MSXDOS no drive. Vamos agora consultar o "índice" do disco, isto é, o diretório.

O comando MSXDOS que permite observar o conteúdo do diretório é o DIR. Para usá-lo, basta comandar:

```
A>DIR
```

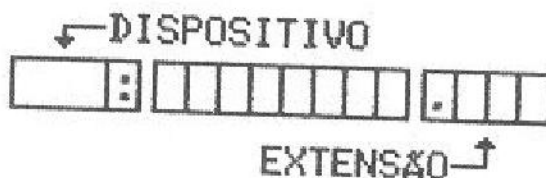
Ao fazer isso, o MSXDOS apresentará na tela a relação de todos os "arquivos" contidos no disco.

Cada um dos itens constantes do diretório é um arquivo. Como você pode observar, eles nada mais são que conjuntos de dados agrupados sob um mesmo nome. Um arquivo pode ser um programa em BASIC, um programa em Linguagem de Máquina, um conjunto de dados qualquer, etc.

Mais adiante, no capítulo 4, aprenderemos mais sobre

os arquivos e seus nomes. Por enquanto, basta saber que todo arquivo em disco tem um nome formado por duas partes, a primeira com até 8 caracteres e a segunda com até três caracteres. O nome pode ter, portanto, até 11 caracteres.

Em alguns casos, é necessário especificar também junto ao nome dos arquivos o drive lógico em que eles se encontram. Por exemplo, um arquivo de nome PROG1.BAS que esteja no drive lógico B: pode ser referenciado como "B:PROG1.BAS". O drive em que o arquivo se encontra pode ser considerado com parte integrante do nome.

FIGURA 3.3 — Estrutura dos nomes dos arquivos em disco.

O comando DIR, além de mostrar os nomes dos arquivos em disco também apresenta seus tamanhos, as datas e as horas marcadas pelo sistema quando eles foram gravados.

CORINGAS

O MSXDOS possui alguns recursos que facilitam em muito a pesquisa de arquivos com nomes semelhantes. Por exemplo, imagine que você queira saber quais os arquivos presentes no disco e que começam com a letra A. Basta comandar:

```
A>DIR A*.*
```

O caractere asterisco (*) é usado como uma espécie de "coringa". O MSXDOS, ao ler o comando acima, procura no disco todos os arquivos que começam com a letra A e que tem quaisquer outros caracteres, tanto na primeira quanto na segunda parte do nome. O caractere asterisco pode ser usado também com outros comandos.

Agora, imagine que você queira procurar no disco os arquivos que começam com a letra A e que tem exatamente 5 caracteres na primeira parte do nome e três caracteres na seguna parte. O asterisco não será muito útil, mas existe um outro "coringa" que o substitui com vantagens neste caso: o caractere interrogação (?). Para executar a procura que acabamos de descrever, basta comandar:

```
A>DIR A????.???
      ¯¯¯¯¯ ¯¯¯
        5    3
```

Note que usamos 4 caracteres interrogação na primeira parte do nome e três na segunda. Cada interrogação substitui um único caractere enquanto que o asterisco substitui uma sequência de caracteres. Ao ler o comando acima o MSXDOS apresentará na tela todos os arquivos procurados. Como o asterisco, o interrogação também pode ser usado com alguns outros comandos.

APRESENTAÇÃO NA TELA

O comando DIR possui duas opções que alteram a forma de apresentação dos dados na tela. Uma delas, a "/W", faz com que apenas os nomes dos arquivos sejam mostrados. Os tamanhos, as datas e as horas são suprimidos da listagem na tela e, ao invés dos nomes serem apresentados em apenas uma coluna, são mostrados em duas ou mais colunas. Com o disco do sistema no drive, experimente comandar:

```
A>DIR /W
```

Você deverá obter uma apresentação do diretório como mostra a figura 3.4.

A outra opção é a "/P" e faz com que a listagem do diretório seja temporariamente interrompida sempre que a tela do vídeo estiver cheia. Isso é muito útil quando estamos usando um disco com muitos arquivos, cujo diretório completo ocupa várias telas.

Quando a listagem do diretório é interrompida devido a opção /P, o MSXDOS pede para que o usuário digite

uma tecla qualquer para continuar.

FIGURA 3.4 — Diretório apresentado com a opção /W.

```
A>DIR /W
MSXDOS   SYS    COMMAND  COM
FUNKEY   BAT    FUNKEY   BAS
ESTATIST BAS    MUSICA   BAS
LISTPRE  BAS    COPYSCR  BAS
COPYSCR  BIN    EDITCAR  BAS
EDITCAR  BIN    BOLD     TAB
      12 files   326656 bytes free
A>█
```

As opções /W e /P podem ser usadas simultaneamente. Por exemplo, pode-se comandar:

```
A>DIR *.COM /W /P
```

Uma outra forma de interferir na apresentação de dados na tela é usando o comando MODE (o primeiro que vimos no capítulo 1). Ele permite limitar o número de colunas do vídeo e é particularmente útil quando a imagem fica parcialmente fora da tela devido à um ajuste deficiente do monitor ou da TV.

Experimente comandar:

```
A>MODE 33
```

Isso fará com que o vídeo fique com apenas 33 colunas.

O número de colunas do vídeo especificado no comando MODE pode ser no máximo 40 (se não se está usando um cartão de 80 colunas, caso em que pode ser até 80).

INTERRUPÇÃO DE COMANDOS

Existe ainda um outro recurso do MSXDOS que nos possibilita parar temporária ou definitivamente a execução de um comando DIR: o uso dos códigos ^S (teclas CONTROL e S pressionadas simultaneamente) e ^C (teclas

CONTROL e C pressionadas simultaneamente).

Ao digitarmos ^S, a listagem é interrompida temporariamente. Para que ela seja retomada, basta digitar novamente ^S. Se, ao invés de ^S digitarmos uma tecla qualquer, o uso, do ^S será desativado e não terá mais ação sobre a listagem do diretório.

A tecla ^C produz uma interrupção mais drástica. Ela pára a listagem defintivamente e devolve o controle ao MSXDOS, apresentando o sinal de pronto (A>) na tela. Ao invés de ^C, pode-se também usar CONTROL e STOP.

Experimente executar um comando DIR e use repetidas vezes ^S para verificar como ele funciona na prática. Experimente também o ^C ou CONTROL e STOP.

ECO NA IMPRESSORA

O MSXDOS permite ainda que todas as informações enviadas para o vídeo sejam enviadas simultaneamente para a impressora. Para isso basta digitar ^P (teclas CONTROL e P pressionadas simultaneamente). Experimente digitar:

```
A>DIR ^P
```

O caractere ^P não aparecerá no vídeo, mas o MSXDOS o reconhecerá e passará a enviar à impressora um "eco" de tudo que for enviado para o vídeo.

Para desativar o "eco" na impressora basta usar o ^N (obtido pressionando-se as teclas CONTROL e N simultaneamente). O caractere ^N também não aparece na tela.

COMO COPIAR OS ARQUIVOS

No capítulo 2 você aprendeu como fazer um BACKUP de seu disco de sistema. Vamos ver com mais detalhes como isso foi feito.

O MSXDOS possui um comando específico para copiar arquivos, o COPY. Para usá-lo basta especificar o nome do arquivo a ser copiado e o nome do arquivo cópia. Por exemplo, suponha que você queira copiar o arquivo DISCO.

BAS para um outro disco. Basta comandar:

```
A>COPY A:DISCO.BAS B:
```

Note que especificamos o nome do drive junto ao nome dos arquivos, pois se o drive B: não fosse especificado a cópia seria feita sobre o próprio arquivo DISCO.BAS.

No exemplo acima o arquivo cópia terá o mesmo nome do arquivo original, isto é, DISCO.BAS. Se quiséssemos que ele tivesse outro nome, bastaria especificá-lo. Por exemplo, se comandarmos:

```
A>COPY A:DISCO.BAS B:UTILITAR.BAS
```

O arquivo cópia será gerado no drive lógico B: e com o nome UTILITAR.BAS.

Agora, vamos ver como usar os coringas com o comando COPY.

Existem casos em que queremos copiar muitos arquivos de um disco, mas não todos. Por exemplo, imagine que você queira copiar todos os arquivos terminados por ".COM". Basta usar o comando COPY da seguinte forma:

```
A>COPY A:*.COM B:
```

Nesse exemplo, o MSXDOS copiaria para o drive lógico B: todos os arquivos terminados por ".COM" e com os mesmos nomes que eles tem no drive lógico A: .

O caractere interrogação pode ser usado com o COPY de forma semelhante. Por exemplo, o comando abaixo faz com que todos os arquivos terminados em ".COM" e de apenas dois caracteres na primeira parte do nome sejam copiados:

```
A>COPY A:??.COM B:
```

INDO PARA O BASIC

Estando em MSXDOS, para irmos para o BASIC basta usar o comando BASIC! Experimente comandar:

```
A>BASIC
```

Ao executar esse comando o MSXDOS se auto-desliga e transfere o comando do micro para o BASIC.

INTRODUÇÃO AO DISK BASIC

PRIMEIRAS INSTRUÇÕES EM DISK BASIC

Quando você liga o micro com a interface conectada (e sem apertar SHIFT ou CONTROL) a primeira coisa que o sistema faz é acionar o drive para ver se há um disco nele. Existindo o disco, serão procurados os seguintes arquivos:

```
MSXDOS  .SYS
```
e
```
COMMAND .COM
```

Quando o MSXDOS encontra esses arquivos, ele reconfigura o MSX deixando-o compatível com inúmeros outros computadores (e perdendo algumas características do MSX).

Se, ao longo desse procedimento, ele não conseguir carregar o MSXDOS, ou porque o acionador está desligado, ou porque não há disco no drive, ou ainda porque o disco não tem os arquivos citados, então será ativado o DISK BASIC MSX, gravado na memória permanente da interface.

Com o DISK BASIC ativado, o MSX conserva suas características peculiares e sua linguagem BASIC original, implementada com alguns recursos adicionais para poder operar com discos.

A primeira mensagem que o DISK BASIC envia para o

usuário é:

Entre data (D-M-A):█

Se você não faz questão de gravar arquivos com a data atual pode digitar simplesmente RETURN. Se você está trabalhando profissionalmente com seu MSX e vai gravar bancos de dados que sofrem atualizações periódicas, é conveniente criar o hábito de sempre digitar a data atual, usando dois dígitos para o Dia, dois para o Mês e dois para o Ano, separados por um traço (sinal de menos). A seguir, digitando RETURN, você terá a tela da figura 4.1.

FIGURA 4.1 — Tela de apresentação do DISK BASIC.

A outra maneira de se acessar o DISK BASIC, como já vimos, é a partir do MSXDOS.

Se você já está com o computador ligado e este carregou o MSXDOS, basta comandar:

A>BASIC (e RETURN)

Dessa forma você obterá a mesma tela apresentada na figura 4.1.

Se você tiver um televisor que apresenta problemas de leitura na primeira ou na última coluna da tela, comande:

```
WIDTH 39
```

Assim você obterá o mesmo efeito que obtinha com o comando MODE 39 do MSXDOS.

A primeira coisa que vamos aprender agora é como ler o diretório do disco estando em BASIC.

Experimente comandar:

FILES (e RETURN)
ou files (e RETURN)

Você obterá na tela a lista dos arquivos contidos no disco. Essa lista é menos completa que a produzida pelo comando DIR do MSXDOS, pois além de não ser fornecida a data de gravação (e eventualmente o horário), não diz quantos bytes cada arquivo ocupa.

O comando FILES aceita ser seguido por nome de arquivos de maneira a listar seletivamente alguns arquivos. Valem, também, os mesmos "coringas" usados no MSXDOS (* e ?).

Lembre-se, porém, que o nome do arquivo (e eventualmente do dispositivo) deve estar ENTRE ASPAS.

Se, por exemplo, você comandar:

```
FILES"B:*.BAS"
```

isso significará que ele listará do drive B (se houver um só acionador será solicitada uma troca de discos) todos os arquivos com a terminação ".BAS".

Se você tiver uma impressora conectada ao seu micro, comande:

LFILES (e RETURN)

Você obterá a listagem dos arquivos contidos no disco na impressora. Com relação aos coringas (* e ?) o LFILES se comporta como o FILES.

A partir do DISK BASIC você também pode formatar discos virgens. Para isso basta comandar:

```
CALL FORMAT
```

Você terá o mesmo menu que aparecia na formatação pelo MSXDOS (figura 2.1).

CÓPIA E BACKUP NO DISK BASIC

O DISK BASIC tem um comando COPY que funciona de maneira análoga ao do MSXDOS. Existe, porém, uma diferença importante: a sintaxe é ligeiramente diferente. Os nomes dos arquivos (e eventualmente dos drives) devem estar entre aspas. Além disso, deve haver um separador "TO" entre o arquivo origem da cópia e o arquivo destino da cópia. Se você quiser copiar, por exemplo, um arquivo de nome "BABA.GUT" presente no drive A para o drive B, renomeando-o para "CACA.NIC", você deverá comandar:

```
COPY "A:BABA.GUT" TO "B:CACA.NIC"
```

Neste caso o sistema lerá o arquivo de nome BABA. GUT no drive A e o gravará como CACA.NIC no drive B.

GRAVAÇÃO E LEITURA DE PROGRAMAS EM BASIC

Quando você tem um programa em BASIC na memória do computador e quer salvá-lo em disco, basta usar o comando SAVE, seguido do nome. Após o nome, de no máximo 8 caracteres, coloque a terminação ".BAS" para se lembrar de que se trata de um programa em BASIC.

Se você resolve chamar seu programa em BASIC, por exemplo, de ALEPH e quer gravá-lo no drive B, basta comandar:

```
SAVE "B:ALEPH.BAS"
```

Se você omitir o nome do drive, o programa será gravado no drive corrente (em binário compactado).

Isso significa que o arquivo no disco será a cópia do conteúdo da RAM (com alguns comandos substituídos por suas "tokens", códigos que permitem compactar as informações de modo a ocupar menos memória).

Em alguns casos, é conveniente copiar o programa "por extenso", sem essa compactação. Então, ele deve ser

gravado em ASCII. O processo é análogo ao do comando LIST, onde cada "token" é transformada na sequência completa de caracteres correspondendo ao comando BASIC. Tudo se passa como se o programa fosse "listado" no disco.

Para gravar em ASCII, basta comandar:

```
SAVE "ALEPH.ASC",A
```

Nesse caso, como não foi especificado o drive, o programa é gravado no drive corrente. O ",A" colocado após o nome do arquivo faz com que a gravação seja feita em ASCII e a terminação ".ASC" serve para que você se lembre, no futuro, de que esse arquivo contém um programa gravado "por extenso".

Os oito caracteres usados no nome devem, obrigatoriamente, ter seus códigos entre 33 e 126 (CHR$(33) e CHR$(126)).

Rode o programa da figura 4.2 para visualizá-los.

FIGURA 4.2 — Caracteres usados nos nomes dos arquivos.

Dos caracteres listados na tela, convém observar que as letras minúsculas são transformadas automaticamente em maiúsculas. Além disso, por serem usados com outras funções, você deve excluir da lista os caracteres:

" * + , . / : ; = ? [\ ]

Os nomes dos dispositivos aceitos pelo sistema também não devem ser usados como nomes de arquivos. Os dispositivos são os seguintes:

AUX — Auxiliar
CON — Teclado (input) ou Vídeo (output)
LST — Impressora
PRN — Impressora
NUL — Nulo

Para recuperar um programa em BASIC gravado no disco, basta comandar:

```
LOAD "A:ALEPH.BAS"
```

ou

```
LOAD "ALEPH.ASC"
```

Você pode omitir o nome do drive (será assumido o drive corrente). Se, após o comando, você colocar um ",R" o programa será carregado e já entrará rodando.

Para fazer uma experiência, grave o programa da figura 4.2 com:

```
SAVE "ALEPH.ASC",A
```

Digite então:

```
LOAD "ALEPH.ASC",R
```

Você verá o micro executar a listagem dos caracteres.
Comande agora:

```
RENUM 100
```

A seguir, comande:

```
LIST
```

Você agora tem, na memória do micro, um programa que vai da linha 100 à 120.
Comande agora:

```
MERGE "ALEPH.ASC"
```

A seguir, comande:

LIST

Você tem agora o programa do disco (das linhas 10 à 30) "emendado" ao programa que estava na memória (das linhas 100 à 120). Este tipo de "emenda" não teria sido possível se o arquivo chamado pelo comando MERGE estivesse gravado em binário compactado.

Gravar um programa em ASCII não tem só esta utilidade. Se você usar um disco de MSX com programas deste tipo e tentar lê-los num micro IBM-PC, terá uma surpresa: eles serão carregados no IBM! A surpresa se torna ainda maior quando você percebe que programas BASIC gravados em ASCII num IBM-PC são lidos e até rodados num MSX! Porém, antes de rodá-los, é conveniente dar uma olhada na listagem, pois apesar dos dois BASIC's serem quase iguais, há algumas instruções que precisam ser eventualmente adaptadas.

Paradoxalmente o BASIC do MSX é mais completo que o do IBM-PC, sendo portanto mais fácil adaptar do PC para o MSX que tentar o caminho contrário.

Se você estiver trabalhando com compiladores, então, a utilidade dos programas gravados em ASCII se torna fundamental.

GRAVAÇÃO E LEITURA DE PROGRAMAS EM LINGUAGEM DE MÁQUINA

Para gravar um programa em Linguagem de Máquina ou dados em binário, deve ser usado o comando BSAVE. Além de usar as mesmas regras para o nome do arquivo, é sempre conveniente usar a terminação ".BIN" para reconhecê-lo como arquivo binário.

Um programa em Linguagem de Máquina ou um arquivo em binário consiste numa sequência de códigos localizados num trecho da memória RAM do micro. Para que o BSAVE saiba qual trecho deve ser copiado em disco, devemos informar o endereço inicial e o final da localização do programa.

Se você comandar, por exemplo:

```
BSAVE "PAG3.BIN",49152,65535
```

todo o conteúdo da página 3 da memória RAM do micro será transferido para o disco no arquivo PAG3.BIN.

Digamos agora que você tenha um jogo em Linguagem de Máquina na memória do micro, localizado entre o endereço 42517 e 47212 e que, para começar a rodar, deva ser inicializado no endereço 43000 com os seguintes comandos:

```
DEFUSR0 = 43000
X = USR0(0)
```

Você pode gravá-lo fornecendo esta informação adicional (endereço de execução) no comando BSAVE. No exemplo dado, o comando a ser digitado seria:

```
BSAVE "JOGO.BIN",42517,47212,43000
```

Para recuperar o arquivo binário, basta usar o comando BLOAD seguido do nome do arquivo. Não tente recuperar o arquivo PAG3.BIN se não você vai produzir uma desorganização catastrófica em seu micro! Aliás, manipular arquivos em binário exige muito cuidado: a "lei de Murphy" tem a máxima prioridade nestes casos!

Digamos que você queira recuperar seu jogo em Linguagem de Máquina. Nesse caso você tem duas opções: ou comanda

```
BLOAD "JOGO.BIN"
```

e depois que a sequência de códigos for carregada na RAM (entre os endereços 42517 e 47212), você comanda

```
DEFUSR0 = 43000
X = USR0(0)
```

para que o jogo comece a rodar, ou então você comanda

```
BLOAD "JOGO.BIN",R
```

Neste caso, como o programa já havia sido gravado com a informação do endereço de execução (43000), ele já entrará rodando.

Se o endereço de execução for omitido durante a gravação com o comando BSAVE e você tentar carregar o

programa com a opção ",R" no final do comando BLOAD, o programa será carregado e começará a ser executado automaticamente a partir de seu endereço inicial.

Digamos agora que o programa que estava na RAM, para rodar, precise ter todos os seus códigos deslocados de 200 posições acima.

Você deverá recuperá-lo com:

```
BLOAD "JOGO.BIN",200
```

Isso produzirá o deslocamento desejado.

COPIANDO TELAS

O comando BSAVE permite copiar no disco não só trechos da RAM, mas também pedaços de VRAM (RAM de vídeo do VDP).

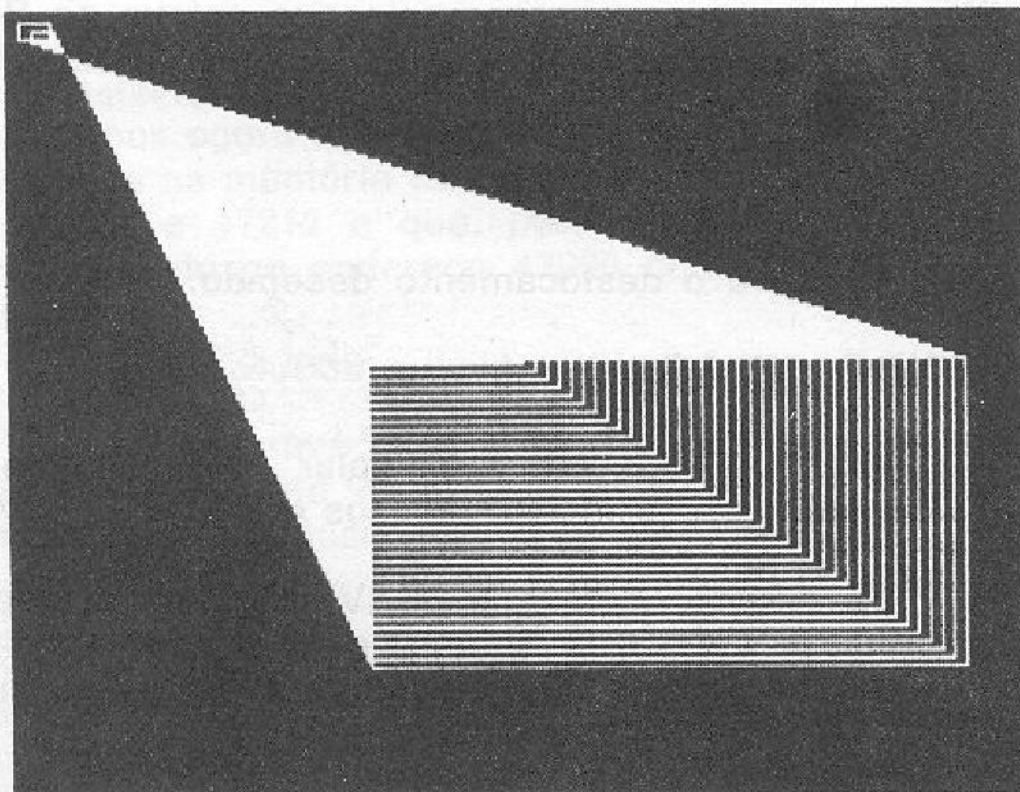
Para isso, após o comando BSAVE você deve colocar o complemento ",S" (S de Screen) para informar o sistema que o arquivo binário deve ser copiado da memória de vídeo.

Digite, por exemplo, o programa da figura 4.3.

FIGURA 4.3 — Gravando uma tela.

```
10 SCREEN 2
20 FOR I=1 TO 80
30 LINE (I+10,I+5)-(3*I,2*I),,B
40 NEXT I
50 IF INKEY$="" THEN GOTO 50
60 BSAVE"TELATEST.SCR",0,16383,S
Ok
```

As linhas de 10 a 40 geram a tela da figura 4.4.

FIGURA 4.4 — Exemplo de tela.

A linha 50 faz o micro aguardar o pressionamento de uma tecla qualquer. Pressione uma tecla (por exemplo, a da letra A) e você verá a luz frontal do acionador se acender, pois começa a ser executado o comando de transferência da tela para o disco:

```
BSAVE "TELATEST.SCR",0,16383,S
```

O nome TELATEST tem 8 caracteres, a terminação SCR é normalmente usada quando o arquivo binário corresponde a uma tela (SCReen), os endereços 0 e 16383 representam o começo e o fim de toda a VRAM (16 Kbytes) e a terminação S informa que estes endereços não são da RAM normal mas sim de RAM de vídeo (VRAM).

Ao terminar a execução da cópia em disco, a luz se apaga e ativa-se no micro a SCREEN 0 para devolver o cursor ao usuário.

Execute um comando:

```
FILES
```

para verificar se seu arquivo TELATEST.SCR foi gravado e a seguir comande um:

NEW

para apagar o programa da figura 4.3 da memória do micro.

Para recuperar a figura, basta digitar o programa da figura 4.5.

FIGURA 4.5 — Recuperando uma tela.

```
10 SCREEN 2
20 BLOAD"TELATEST.SCR",S
30 GOTO 30
Ok
```

Ao rodar este programa você verá a luz de acesso se acender e, a seguir, a tela da figura 4.4 aparecer instantaneamente no vídeo.

Vamos agora fazer uma última experiência para resumir tudo que vimos até agora. Comande:

NEW

para limpar o micro (antes você deve ter dado CONTROL + STOP para recuperar o cursor) e digite o programa da figura 4.6.

FIGURA 4.6 — Recuperando "penosamente" uma tela.

```
10 BLOAD"TELATEST.SCR",33000!
20 PRINT"PRESSIONE UMA TECLA"
30 IF INKEY$="" THEN GOTO 30
40 SCREEN 2
50 FOR I=0 TO 16383
60 VPOKE I,PEEK(I+33000!)
70 NEXT I
80 GOTO 80
Ok
```

Comande:

RUN

Digite uma tecla quando solicitado e veja o programa funcionando. Como ele vai demorar um certo tempo, enquanto espera algum efeito, leia a explicação a seguir:

A linha 10 carrega os 16 Kbytes da tela na RAM normal, deslocando os endereços de 33.000 posições para cima do endereço 0. A seguir o micro solicita o pressionamento de uma tecla para passar à fase seguinte.

Neste ponto ele busca sequencialmente os 16 Kbytes que estão na RAM, com a função:

```
PEEK (I+33000!)
```

e os coloca na VRAM com o comando:

```
VPOKE
```

Ao fazer isso, a tela vai sendo penosamente construída, byte a byte, até se completar.

Esse seria o processo que você deveria usar se os comandos BSAVE e BLOAD não tivessem o poderoso recurso do ",S"!

Se você ler o livro "APROFUNDANDO-SE NO MSX" para estudar a organização da VRAM, poderá gerar recursos incríveis.

Só como exemplo, você pode ter uma tela A com alguns SPRITES e uma tela B sem SPRITES. Usando o recurso do ",S" e sabendo os endereços certos você pode copiar a tela B com os SPRITES da tela A!

VOLTANDO DO BASIC PARA O MSXDOS

Vamos agora deixar o DISK BASIC e retornar ao MSXDOS. O comando do BASIC que transfere o controle do sistema para o MSXDOS é o CALL SYSTEM. Antes de usá-lo, insira no drive um disco que possua os arquivos MSXDOS.SYS e COMMAND.COM. Feito isso, comande:

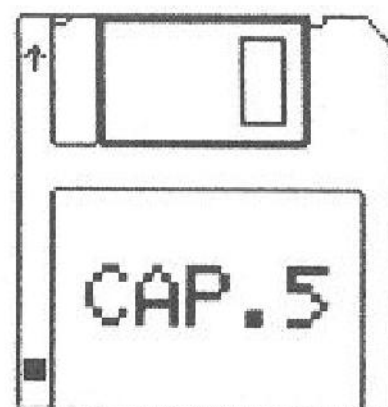
```
CALL SYSTEM
```

ou

```
_SYSTEM
```

Com isso passaremos a operar em MSXDOS e o "A>" estará de volta a tela.

Esse retorno ao MSXDOS só é possível se você chegou ao BASIC pelo comando: `A>BASIC`

USANDO ARQUIVOS EM MSXDOS

TIPOS DE ARQUIVOS

Um disco pode conter muitos tipos de informações armazenadas na forma de arquivos. É comum, e o uso tem mostrado também que é extremamente útil, usar os três últimos caracteres do nome para especificar o tipo de informação contida num arquivo.

Num mesmo disco podem co-existir arquivos de programas e arquivos de dados.

Os arquivos de programas são sequências de instruções escritas em alguma linguagem de programação. Por exemplo, BASIC, ALGOL, FORTRAN, COBOL, PASCAL, C, FORTH, ASSEMBLY, LISP, ADA, MUMPS, PROLOG LOGO, etc.

Os arquivos de dados são apenas sequências de caracteres que podem ser usados por alguns programa, como um gerenciador de banco de dados ou um editor de textos.

A forma de gravação também é uma característica que pode ser indicada pela terminação do nome. Arquivos gravados pelo DISK BASIC, por exemplo, podem diferir de arquivos gravados por outros programas que são executados através do MSXDOS.

A seguir relacionamos algumas terminações consagradas pelo uso entre os usuários de microcomputadores.

TERMINAÇÃO	TIPO DE ARQUIVO
.BAS	programas-fonte em BASIC.
.SCR	arquivos de telas gravados pelo Disk Basic.
.BIN	arquivos gravados em binário, normalmente são programas em Linguagem de Máquina que rodam a partir do Disk Basic.
.ASM	programas-fonte em ASSEMBLY.
.ASS	programas-fonte em ASSEMBLY.
.COB	programas-fonte em COBOL.
.PAS	programas-fonte em PASCAL.
.DAT	dados diversos.
.TXT	textos diversos.
.COM	programas em Linguagem de Máquina que rodam sob o MSXDOS.
.BAT	arquivos de comandos DOS.
.ASC	arquivos gravados em ASCII.
.BAK	arquivos cópias de outros.
.BKP	arquivos cópias de outros.
.GAM	arquivos de jogos.

Habitue-se a usar sempre as terminações nos nomes dos seus arquivos. Enquanto você tiver poucos discos, isso pode lhe parecer desnecessário, mas quando sua "discoteca" passar dos dez discos repletos de programas, o uso das terminações tornará a procura de arquivos extremamente mais rápida! Os outros 8 caracteres dos nomes dos arquivos, como vimos no capítulo 4, podem conter quaisquer caracteres de códigos ASCII entre 33 e 126, com exceção dos 13 seguintes:

" * + , . / : ; = ? [\ ]

Lembre-se de não usar como nome de arquivo nenhum nome de dispositivo: AUX, CON, LST, PRN e NUL.

LISTANDO O CONTEÚDO DE ARQUIVOS

O MSXDOS possui um comando que nos permite listar na tela o conteúdo dos arquivos. Obviamente, muitos arquivos não farão o mínimo sentido para nós, pois são

constituídos de códigos a serem usados por algum programa ou pelo próprio computador. Outros arquivos, entretanto, podem ser lidos e compreendidos por nós. Por exemplo, os arquivos de programas-fonte, os arquivos de texto e alguns arquivos de dados.

O comando que permite listar arquivos na tela é o TYPE e para usá-lo basta especificar o nome do arquivo à sua frente. Por exemplo, para listar o conteúdo do arquivo DISCO.ASC na tela, basta comandar:

```
A>TYPE DISCO.ASC
```

Para interromper a listagem você pode usar ^S ou ^C. Caso você tenha uma impressora conectada ao seu micro, o "eco" poderá ser ativado ou desativado com ^P e ^N, respectivamente.

APAGANDO ARQUIVOS DO DISCO

O MSXDOS possui um comando que nos permite eliminar arquivos gravados num disco, desde que ele não esteja com o dispositivo contra gravação acionado, isto é, a janela da trava contra gravação deve estar fechada!

Esse comando pode ser acessado por dois nomes diferentes, porém em qualquer caso sua função será a mesma. Os nomes são ERASE ou DEL.

Por exemplo, suponha que você queira apagar do disco o arquivo DISCO.BKP .Para isso, basta comandar:

```
A>ERASE DISCO.BKP
```

ou

```
A>DEL DISCO.BKP
```

O comando ERASE também permite o uso dos coringas.

Suponha que você queira apagar do disco todos os arquivos de cópias, ou seja, os terminados por ".BKP". Basta usar o comando ERASE da seguinte forma:

```
A>ERASE *.BKP
```

O caractere interrogação pode ser usado da mesma forma e com a mesma função que no comando DIR.

Um outro recurso muito útil do MSXDOS é o comando RENAME ou REN. Ele permite renomear os arquivos de um disco. Para usá-lo basta especificar o nome atual do arquivo e o novo nome que ele deverá receber. Por exemplo, se você quiser alterar um arquivo de nome DISCO.BKP para DISCOUTI.BKP, bastará comandar:

```
A>RENAME DISCO.BKP DISCOUTI.BKP
```

O RENAME também pode ser usado com os coringas. Por exemplo, para alterar o nome de todos os arquivos de programas-fonte em BASIC do disco de modo que eles comecem sempre com a letra "X" basta comandar:

```
A>RENAME *.BAS X*.*
```

Se o novo nome atribuído a um arquivo já existir no disco, o MSXDOS interromperá a execução do comando RENAME e apresentará uma mensagem de erro na tela.

ARQUIVOS DE COMANDOS MSXDOS

Um dos tipos de arquivos que podemos ter em discos são os de comandos do próprio MSXDOS, chamados de arquivos BATCH (lote, em inglês). Esses arquivos devem necessariamente ser terminados por ".BAT", pois caso contrário não serão executados pelo sistema.

Um arquivo BATCH pode conter quaisquer comandos do MSXDOS. Para gerar um arquivo BATCH deve-se usar um editor de textos (como o MSX-WRITE, o SCED ou o WORDSTAR) ou o próprio comando COPY do MSXDOS. Obviamente, um editor de textos oferece inúmeras vantagens em relação ao COPY mas não é necessário para a maioria das aplicações.

Vamos criar um arquivo BATCH que contenha o comando MODE 38 e, logo a seguir, o comando DIR. Para isso, devemos usar a seguinte construção do comando COPY:

```
A>COPY CON A:TESTE.BAT
```

Note que usamos o CONsole como dispositivo de entrada (TECLADO) e o drive A: como dispositivo de saída.

Isto significa que o arquivo a ser copiado deverá ser digitado pelo teclado e será gravado no drive A: com o nome "TESTE.BAT".

Após receber o comando acima, o MSXDOS estará a espera do texto a ser digitado no teclado. Digite, então:

```
A>MODE 38                 <e RETURN>
```

e

```
A>DIR                     <e RETURN>
```

Para informar que o texto terminou, devemos digitar o caractere correspondente a "fim de arquivo", ou seja, ^Z (CONTROL + Z). Digite, então, ^Z. A tela deverá estar mais ou menos como a figura 5.1.

FIGURA 5.1 — Uso do COPY para gerar arquivos BATCH.

```
A>
A>
A>COPY CON A:TESTE.BAT
MODE 38
DIR
^Z
        1 file copied
A>
```

Uma vez com o arquivo BATCH gravado no disco, podemos executá-lo simplesmente digitando seu nome SEM A TERMINAÇÃO. Experimente comandar:

```
A>TESTE
```

Ao pressionar RETURN, o MSXDOS vai procurar no disco algum arquivo com o nome TESTE.BAT (ou TESTE.COM) e, achando-o, o executará. Note que, se existirem num mesmo disco dois arquivos, um terminado em ".COM" e outro em ".BAT", e com os oito primeiros caracteres do nome coincidentes, o MSXDOS dará prioridade ao terminado em ".COM". Portanto, cuidado para não nomear arquivos BATCH com nomes de arquivos ".COM"!

Após a execução do arquivo TESTE.BAT a tela deverá apresentar o diretório do disco em uso (figura 5.2).

FIGURA 5.2 — Execução do arquivo TESTE.BAT.

```
A>DIR
MSXDOS   SYS      2560 20-11-87   9:07a
COMMAND  COM      6656 20-11-87   7:56a
FUNKEY   BAT        19 10-11-87
FUNKEY   BAS      1745 10-11-87
ESTATIST BAS      2408 10-11-87
MUSICA   BAS      2678 10-11-87
LISTPRE  BAS      1482 10-11-87
COPYSCR  BAS      2005 10-11-87
COPYSCR  BIN       608 10-11-87
EDITCAR  BAS      4460 10-11-87
EDITCAR  BIN      2807 10-11-87
BOLD     TAB      2056 10-11-87
TESTE    BAT        15  1-01-84
        13 files   325632 bytes free
A>
A>
```

Os arquivos BATCH permitem o uso de diversos outros recursos. Vamos aprender dois novos comandos do MSXDOS que imprimem mensagens na tela e que tem seu uso mais frequente nos arquivos BATCH: o REM e o PAUSE.

O comando REM apenas mostra na tela a mensagem que estiver à sua frente. Para experimentá-lo, vamos refazer o arquivo TESTE.BAT. Digite os comandos a seguir:

```
A>COPY CON A:TESTE.BAT
MODE 38
REM==================================
REM== DIRETóRIO DO DRIVE CORRENTE ==
REM==================================
DIR
^Z
```

Agora, comande:

```
A>TESTE
```

As mensagens dos comandos REM deverão surgir na tela antes que o diretório do disco seja apresentado.

O comando PAUSE, além de apresentar mensagens na tela de modo idêntico ao comando REM, interrompe a execução do arquivo BATCH até que alguma tecla seja pressionada pelo usuário. Isso pode ser particularmente útil quando temos mais mensagens do que cabem numa única tela. Podemos, nesse caso, usar o comando PAUSE para que o usuário, após ler as mensagens apresentadas na primeira tela, possa ler a(s) tela(s) seguinte(s). Vamos redigitar o arquivo TESTE.BAT usando o comando PAUSE.

```
A>COPY CON A:TESTE.BAT
MODE 38
REM===================================
REM== DIRETÓRIO DO DRIVE CORRENTE ==
REM===================================
REM    PARA VER O DIRETÓRIO
PAUSE   PRESSIONE ALGUMA TECLA
^Z
```

EDIÇÃO DE LINHAS DE COMANDOS

Durante a digitação de uma linha de comando em MSXDOS podem ocorrer erros, ou então, você pode ter que repetir a maior parte de uma linha já digitada. O MSXDOS possui diversos recursos para edição das linhas de comandos.

Vamos ver rapidamente cada um deles.

Digite DIR /W/P para usarmos como exemplo de uma linha de comando.

Após a execução do comando, experimente usar as teclas de setas.

Então? Percebeu as funções de edição que elas possuem? Essas teclas são de grande utilidade e existem ainda algumas outras com funções mais complexas.

Experimente digitar um outro comando, por exemplo, DIR *.BAS /W/P, e antes de pressionar RETURN, digite ^K (CONTROL e K). A linha não foi executada, mas está na área de edição e pode ser alterada.

Para perceber melhor a utilidade desse recurso, tente fazer a mesma coisa sem usar o ^K!

A seguir encontram-se resumidos os recursos de edição do MSXDOS. Experimente usá-los com alguns comandos para compreender melhor como eles funcionam na prática.

▶ ou ^ (SETA PARA A DIREITA)
Copia um caractere da área de edição para a tela e desloca o cursor para a direita.
◀ ou ^H ou ^] ou BS (SETA PARA A ESQUERDA)
Apaga da tela o caractere à esquerda do cursor e o desloca uma posição para a esquerda.
▼ ou ^— (SETA PARA BAIXO)
Copia todos os caracteres da área de edição para a tela e posiciona o cursor no fim da linha.
▲ ou ESC OU ^^ ou ^U ou ^Ç (SETA PARA CIMA)
Apaga da tela a linha de comandos em que está o cursor.
DELETE
Apaga da tela o caractere sob o cursor e desloca o restante da linha uma posição para a esquerda.
INSERT ou ^R
Entra e sai do modo de inserção.
HOME ou ^K
Copia a linha em que está o cursor na tela para a área de edição.
SELECTcaractere ou ^Xcaractere
Copia todos os caracteres da área de edição para a tela até achar o caractere.
CLScaractere ou ^Lcaractere
Copia para a tela os caracteres da área de edição a partir da primeira ocorrência do caractere.

AUTO-EXECUÇÃO DE COMANDOS

Logo após o equipamento ser ligado, se existir o MSXDOS no disco presente no drive, ele passará a controlar o equipamento.

Assim que assume o controle do sistema, o MSXDOS procura no disco um arquivo com um nome muito especial: AUTOEXEC.BAT.

Se esse arquivo existir, o MSXDOS tentará executá-lo, assumindo-o como um arquivo BATCH. Portanto, todo arquivo com o nome AUTOEXEC.BAT deve ser necessariamente um arquivo BATCH.

A utilidade deste tipo de arquivo auto-executável fica-

rá mais clara após analisarmos o exemplo a seguir. Reproduza os comandos mostrados na figura 5.3 de modo a criar um arquivo AUTOEXEC.BAT num disco com o MSXDOS.

FIGURA 5.3 — Criação de um AUTOEXEC.BAT.

```
A>COPY CON A:AUTOEXEC.BAT
MODE 38
REM ======================================
REM ========= AUTOEXEC.BAT ==========
REM ======================================
DIR /W /P
PAUSE
REM ======================================
REM ======== ARQUIVOS ".COM" ========
REM ======================================
DIR *.COM
PAUSE
REM ======================================
REM ======== ARQUIVOS ".BAT" ========
REM ======================================
DIR *.BAT
PAUSE
REM ======================================
REM ====== FIM DO AUTOEXEC.BAT ======
^Z
        1 file copied
A>
```

Após ter gerado o arquivo AUTOEXEC.BAT num disco com o MSXDOS, reinicialize o sistema, produzindo um RESET ou desligando o micro e tornando a ligá-lo.

Se tudo correu bem, o MSXDOS deverá assumir o controle do sistema e executará o AUTOEXEC.BAT, mostrando na tela o diretório, os arquivos ".COM" e ".BAT" e as mensagens dos comandos REM.

Experimente criar outros arquivos auto-executáveis, mas lembre-se que só pode existir um em cada disco (pois o nome é sempre AUTOEXEC.BAT) e que eles só funcionarão se o MSXDOS estiver presente.

ENTRADA DE PARÂMETROS PARA ARQUIVOS BATCH

Antes de mais nada, digite os comandos a seguir e crie o arquivo PARAMETRO.BAT.

```
A>COPY CON A:PARAMETRO.BAT
MODE 39
REM====================================
REM== USO DE PARAMETROS EM ARQUIVOS ==
REM== DE COMANDOS DO MSXDOS  (.BAT) ==
REM====================================
REM PARAMETRO 0 = %0
REM PARAMETRO 1 = %1
REM PARAMETRO 2 = %2
REM PARAMETRO 3 = %3
PAUSE
^Z
```

Após gerar esse arquivo, execute-o digitando a linha abaixo:

```
A>PARAMETRO EDITORA ALEPH 1987
```

O MSXDOS permite que sejam usados até 10 (dez) parâmetros genéricos dentro de arquivos BAT. Cada parâmetro é indicado pelo símbolo de porcentagem (%) seguido de um número entre 0 e 9.

O parâmetro %0 será sempre substituido pelo próprio nome do arquivo BATCH em que ele está.

Os demais parâmetros deverão ser fornecidos pelo usuário quando for digitada a linha de execução do arquivo BATCH. No exemplo anterior, as palavras EDITORA e ALEPH e o número 1987 serão atribuídos respectivamente aos parâmetros %1, %2 e %3.

Este recurso do MSXDOS é particularmente útil para a manipulação de arquivos durante compilações de múltiplos passos. Se você for utilizar linguagens compiladas entenderá melhor o que acabamos de dizer.

ATUALIZAÇÃO DA DATA E DO INSTANTE

Vamos ver mais dois comandos do MSXDOS relacionados com o instante em que o sistema está sendo usado: o DATE e o TIME.

Sempre que o MSXDOS é inicializado, ele pede a introdução da data atual. Essa data deve ser fornecida na forma:

`dd/mm/aa` ou `dd-mm-aa`

Sendo dd o dia, mm o mês e aa o ano a serem assumidos pelo sistema. Se digitamos apenas RETURN como resposta, ele assume como data atual a última data corrente fornecida pelo sistema (micros com relógio) ou a data pré-gravada no próprio DOS (micros sem relógio).

O comando DATE permite que você possa consultar ou atualizar a data corrente do sistema. Se o seu MSX possui relógio interno (ou em cartucho), basta acertar a data uma única vez, pois o sistema a atualizará por vários anos automaticamente.

Experimente comandar simplesmente:

```
A>DATE
```

O MSXDOS deverá mostrar na tela a data corrente do sistema. Vamos alterá-la. Digite RETURN, e a seguir, o comando:

```
A>DATE 16-11-1987
```

Agora, para verificar se realmente essa data foi assumida pelo MSXDOS, comande mais uma vez:

```
A>DATE          (e duas vezes RETURN !)
```

O comando DATE é muito útil quando se trabalha profissionalmente com o MSXDOS e mesmo usuários menos experientes podem precisar dele para algumas aplicações.

O comando TIME só funciona se existir relógio no sistema e permite acertar o instante corrente. Após ser ajustado uma vez, ele também não precisa mais ser usado.

MAIS RECURSOS DO COMANDO COPY

O comando COPY possui muitos outros recursos de uso um pouco mais complexos que os já vistos até agora. Conhecendo-os, você poderá juntar arquivos ou transferí-los para outros dispositivos além dos drives: NUL, CON, PRN ou LST, e AUX. Nenhum dispositivo deve ser usado como nome de arquivo! Tome cuidado!

NUL

Este nome de dispositivo deve ser usado quando um

comando requer, como parâmetro, um nome de arquivo, mas não se quer efetivamente criá-lo em disco ou apresentá-lo no vídeo ou na impressora. Alguns compiladores geram automaticamente arquivos com informações adicionais sobre a própria compilação. Se você não quiser que eles sejam gerados em disco e nem quer perder tempo esperando que sejam listados no vídeo ou na impressora, basta direcionar a saída desses arquivos para o dispositivo NUL.

AUX

Este nome de dispositivo deve ser usado para nos referirmos a um dispositivo AUXiliar, como por exemplo uma impressora ou mesmo um drive. O dispositivo AUX deve ser configurado por algum software.

CON

O CONsole, quando usado como dispositivo de entrada de dados, corresponde ao TECLADO. Quando usado como saída de dados, corresponde ao VÍDEO.

PRN ou LST

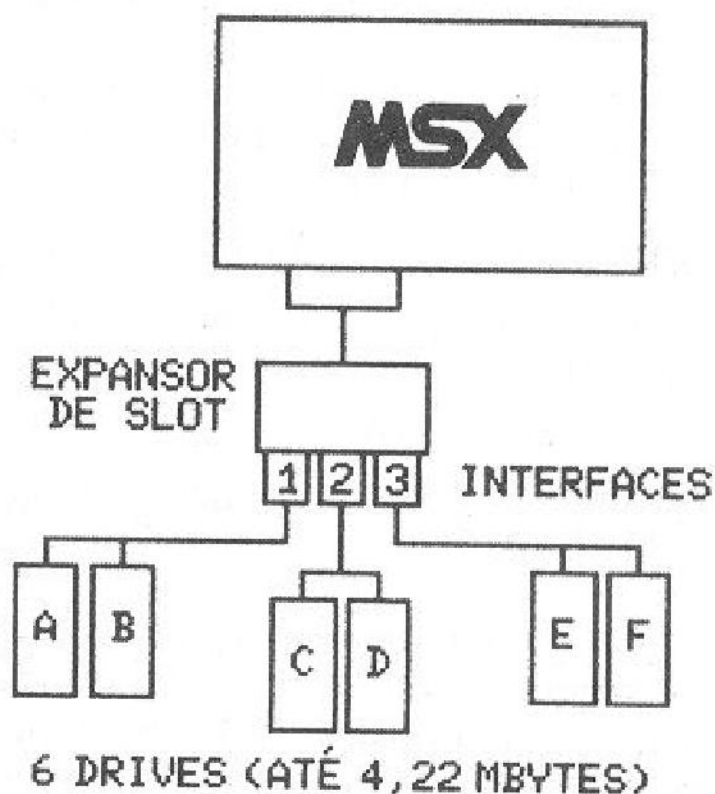
É a impressora.

A:, B:, C:, D:, E: e F:

Os micros MSX permitem a conexão simultânea de até 6 acionadores, sendo necessárias 3 interfaces para controlá-los. Como só existem dois slots disponíveis, normalmente podemos ligar apenas duas interfaces (ou seja, 4 drives). Esse problema, entretanto, pode ser resolvido com um expansor de slots. A configuração com 6 acionadores de 80 trilhas (DS), é uma das maiores possíveis e, nesse caso, teremos uma capacidade disponível de armazenamento por volta de 6 x 720 Kbytes, ou seja, 4,22 Mbytes !!! Essa capacidade só é superada pelo uso de

acionadores de disco rígido, que armazenam normalmente mais de 10 Mbytes cada!

A configuração com seis drives é ilustrada na figura 5.4.

FIGURA 5.4 — MSX com 6 acionadores de disco conectados.

Já que estamos falando de capacidade de armazenamento, vale a pena lembrar que o MSXDOS é um dos únicos sistemas capaz de gerenciar arquivos com mais de 40 Mbytes! A título de comparação, basta lembrar que o CP/M (um dos DOS mais "badalados" pelos programadores profissionais) consegue gerenciar arquivos de no máximo 8 Mbytes.

O comando COPY permite enviar cópias de arquivos de um dispositivo para outro. Além disso, é possível também concaternar arquivos, isto é, juntar vários arquivos num só durante uma cópia. Vamos testar a concatenação, antes porém, gere os dois arquivos BATCH com os comandos a seguir:

```
A>COPY CON A:CLS.BAT
MODE 39
^Z

A>COPY CON A:BEEP.BAT
^G^G^G
^Z
```

Agora você deve ter dois arquivos BAT a mais no seu disco. O primeiro, ao ser executado, apenas produz três "beeps"; o segundo, limpa a tela através de um comando MODE 39.

Vamos gerar um terceiro arquivo que faça as duas coisas" primeiro limpe a tela e a seguir gere os três "beeps". Digite o comando a seguir:

```
A>COPY CLS.BAT + BEEP.BAT = CLSBEEP.BAT
```

Agora liste o diretório na tela para ver se o novo arquivo foi realmente gerado e liste-o com o comando:

```
A>TYPE A:CLSBEEP.BAT
```

Experimente usar esse novo arquivo digitando:

```
A>CLSBEEP
```

Observe que bastou usar o sinal de soma (+) entre os arquivos a serem "emendados" para produzir a concatenação. O sinal de igualdade (=) não é necessário, mas é conveniente usá-lo para não fazer confusão. Se você preferir, não o use, substituindo-o apenas pelo espaço em branco ou por um dos outros separadores aceitos pelo MSXDOS (figura 5.5.).

FIGURA 5.5 — Os separadores do MSXDOS.

```
TAB SPACE , ; =
```

Você pode concatenar mais de dois arquivos numa mesma cópia digitando cada um de seus nomes separados pelo sinal de soma (+) ou, se for mais fácil, usando os coringas (* e ?).

Apenas a título de ilustração, digite o seguinte comando:

```
A>COPY A:*.BAT = A:BATCHS.BAT
```

Ao ser executado esse comando gera o arquivo BATCHS.BAT como sendo a concatenação de todos os arquivos BATCH presentes no drive A (os terminados por ".BAT").

Experimente listá-lo na tela com um comando TYPE e depois apague-o do disco com o comando ERASE.

Até agora usamos exemplos apenas de arquivos BATCH, entretanto podemos emendar outros tipos de arquivos (textos, por exemplo). Para isso, entretanto, são necessários alguns cuidados especiais, pois o comando COPY normalmente interpreta o caractere ^Z como "fim de arquivo" e não o concatena. Se você for emendar programas em linguagem de máquina (terminados por ".COM") será necessário avisar ao COPY que mesmo que ele ache algum caractere ^Z dentro dos arquivos, deve copiá-lo. Para isso, basta usar a opção "/B" (de Binário) no comando COPY. Por exemplo, suponha que você queira emendar o arquivo MSXDOS.SYS com o arquivo COMMAND.COM (apesar de isso não servir para nada!). Bastará comandar:

```
A>COPY/B MSXDOS.SYS+COMMAND.COM=LIXO
```

Se você gerou o arquivo LIXO, apague-o com o comando ERASE.

Para que o comando COPY volte a funcionar normalmente, deve-se usar a opção "/A" (de Ascii) da mesma forma que usamos a opção "/B".

Você pode também usar as opções "/B" e "/A" mais de uma vez na mesma instrução COPY.

Uma última opção do comando COPY é essencial se você quiser aumentar a segurança no processo de transferência de arquivos entre dispositivos: a opção "/V", de "Verificar".

Essa opção faz com que o comando COPY verifique se o conteúdo dos arquivos cópias está igual ao dos arquivos originais. Se for detectada alguma diferença, o MSXDOS enviará uma mensagem ao usuário avisando-o que houve um erro de gravação e solicitando uma ordem para "Abortar" a cópia, "Ignorar" o erro ou "Reentrar" novamente. Para usar a opção "/V" basta incluí-la ao final da linha de comando. Veja no exemplo a seguir o uso correto da

opção "/V".

```
A>COPY A:*.* B:/V
```

Esse comando faz com que todos os arquivos do drive A sejam copiados para o drive B com verificação da gravação. Obviamente, o comando COPY se torna um pouco mais lento com o uso dessa opção, porém esse é um "preço" baixo a ser pago pelo aumento na confiabilidade das cópias.

A opção "/V" só atua no comando em que ela for incluída. Para fazer com que todos os comandos COPY façam automaticamente verificação de gravação, devemos usar um outro comando: o VERIFY.

Esse comando pode ser usado com duas opções: "ON" para ligar a verificação e "OFF" para desligá-la.

Por exemplo, se você comandar:

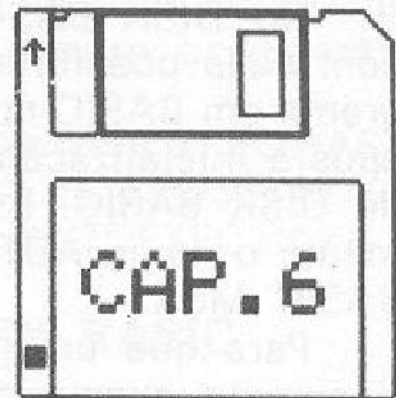
```
A>VERIFY ON
```

O MSXDOS fará com que todos os comandos COPY usados a partir de então executem automaticamente a verificação das gravações.

Para desligar a verificação automática, deve-se comandar:

```
A>VERIFY OFF
```

O que vimos até aqui encerra os principais recursos do MSXDOS. Você poderá encontrar alguns outros detalhes sobre a sintaxe dos comandos no capítulo 7.

PROGRAMANDO COM O DISK BASIC

O DISCO SEM O MSXDOS

Vimos, no capítulo 3 que quando estamos usando o MSXDOS, podemos "sair" dele e passar a usar o DISK BASIC. Podemos também usar o DISK BASIC sem passar pelo MSXDOS.

Experimente inserir um disco virgem no drive e produza um "reset" do sistema, ou desligue-o e torne a ligá-lo em seguida. Você verá que o DISK BASIC assumirá o controle, pedindo a atualização da data. Não retire o disco do drive, pois vamos usá-lo mais adiante. O DISK BASIC possui muito mais recursos que o MSXDOS e é natural que seja um pouco mais difícil dominá-lo.

No capítulo 8 você encontrará uma relação de todos os comandos específicos do DISK BASIC. Neste capítulo, vamos aprender alguma coisa sobre a programação usando os recursos do disco.

Antes de seguir adiante, lembre-se dos comandos já vistos no capítulo 3: FILES, SAVE, LOAD, BSAVE, BLOAD, COPY, CALL FORMAT e CALL SYSTEM.

O PROGRAMA AUTO-EXECUTÁVEL EM DISK BASIC

Vimos que no MSXDOS podem ser criados arquivos de comandos auto-executáveis, os arquivos BATCH de nome AUTOEXEC.BAT.

No DISK BASIC temos algo muito parecido, porém com mais possibilidades. Podemos criar no disco um programa em BASIC que seja automaticamente executado logo após a inicialização do sistema, desde que o controle seja do DISK BASIC. Esses programas devem ter necessariamente o nome AUTOEXEC.BAS e devem estar escritos em BASIC MSX.

Para que um programa AUTOEXEC.BAS seja automaticamente executado, o disco em que ele está não deve ter o MSXDOS, pois nesse caso o DISK BASIC não assumirá o controle logo após a inicialização do sistema.

Formate o disco virgem e destravado, que deve estar no drive, com o comando:

```
CALL FORMAT
```

Escolha a opção 3 (80 trilhas, face simples) se o seu drive for o DT-300 e a opção 4 (80 trilhas, face dupla) se o seu drive for o DT-350.

Após ter formatado o disco, vamos gerar um programa e gravá-lo com o nome AUTOEXEC.BAS.

Não grave o MSXDOS.SYS nesse disco para podermos trabalhar apenas em DISK BASIC.

Digite o programa da figura 6.1:

FIGURA 6.1 — Programa AUTOEXEC.BAS.

```
10 REM
20 REM Programa AUTOEXEC.BAS
30 REM
40 SCREEN 0: WIDTH 38: KEY OFF: PRINT
50 PRINT"ARQUIVOS DESTE DISCO:"
60 PRINT: PRINT: FILES
70 PRINT
80 END
```

Agora, grave-o no disco com o comando:

```
SAVE "AUTOEXEC.BAS"
```

Para certificar-se que o arquivo foi realmente gerado no disco, use o comando:

```
FILES
```

Se tudo correu bem, experimente ressetar novamente

o sistema, sem tirar o disco do drive. O DISK BASIC deverá assumir o controle e automaticamente executará o programa AUTOEXEC.BAS.

Crie você mesmo outros programas AUTOEXEC.BAS e grave-os em seus discos personalizando-os. Lembre-se que cada disco pode ter um único arquivo AUTOEXEC.BAS!

ARQUIVOS GERADOS PELO DISK BASIC

Uma das grandes vantagens dos discos sobre as fitas magnéticas é a velocidade e a confiabilidade das gravações e leituras de dados. Além de armazenar programas em arquivos no disco, podemos também criar arquivos de dados diversos. Por exemplo, podemos criar a partir do DISK BASIC arquivos de dados sobre sua biblioteca ou discoteca particular.

Esses arquivos podem ser de dois tipos diferentes: sequenciais ou randômicos (também chamados de aleatórios). O que distingue um tipo de arquivo do outro é a estrutura em que os dados são armazenados. Por exemplo, imagine uma lista bem simples que relacione nomes, em sua primeira coluna, com telefones, em sua segunda coluna:

```
ANA 63-6205
SUZY 544-6302
CARLA (021) 298-3202
```

Note que tanto os nomes quanto os telefones podem ter tamanhos diferentes.

Num arquivo SEQUENCIAL esses dados estariam gravados como mostra a figura 6.2.

FIGURA 6.2 — Lista em arquivo sequencial.

```
|A|N|A| |6|3|-|6|2|0|5|♪|▣|
|S|U|Z|Y| |5|4|4|-|6|3|0|2|♪|▣|
|C|A|R|L|A| |(|0|2|1|)|2|9|8|-|3|2|0|2|♪|▣|└|

♪ = CHR$ (13) = CR  } SEPARADOR
▣ = CHR$ (10) = LF  }
└ = CHR$ (26) = SUB} FIM DE ARQUIVO
```

Note que, entre os dados existe um separador, formado normalmente pelos caracteres de códigos ASCII 13 (CR = Carriage Return) e 10 (LF = Line Feed), para indicar ao DISK BASIC que alí termina uma informação e começa outra e no final do arquivo existe outro sinalizador (chamado normalmente de SUBstitute character), de código ASCII 26, para indicar o término dos dados.

Num arquivo randômico, essa mesma lista estaria armazenada de outra forma, como mostra a figura 6.3.

FIGURA 6.3 — Lista em arquivo aleatório.

A	N	A		6	3	-	6	2	0	5								
S	U	Z	Y		5	4	4	-	6	3	0	2						
C	A	R	L	A		(	0	2	1	)	2	9	8	-	3	2	0	2

O arquivo randômico (ou aleatório) não necessita de separadores e do indicador de fim de arquivo, porém as informações devem ser de tamanho fixo. Cada informação é chamada de REGISTRO (ou ficha) e pode conter várias partes chamadas CAMPOS. As posições não usadas por dados mais curtos que os campos são preenchidas com espaços em branco até que eles fiquem com o mesmo tamanho dos dados mais longos.

Cada tipo de arquivo tem suas utilidades, suas vantagens e desvantagens. Com o uso você aprenderá como utilizar cada um deles quando for mais conveniente.

Como regra geral, podemos dizer que os arquivos sequenciais (que normalmente consomem menos memória) devem ser usados quando os dados que eles contêm não necessitam ser acessados individualmente, enquanto os arquivos randômicos se prestam justamente para esses casos.

ARQUIVOS DE ACESSO SEQUENCIAL

Um arquivo de acesso sequencial pode ser aberto de três diferentes modos:

1 — Como saída (OUTPUT) de dados.

Nesse caso, o arquivo é criado no disco para

receber dados enviados pelo micro. Se o arquivo já existir no disco, seus antigos dados serão perdidos, pois ele será criado novamente. Não é possível ler dados, mas apenas gravá-los. A sequência de uso nesse caso é a seguinte:

modo OUTPUT [
— Criar e abrir o arquivo;
— Gravar dados no arquivo;
— Fechar o arquivo.

2 — Como saída (APPEND) de dados.
Nesse caso, o arquivo já deve existir no disco (portanto, não será criado) e receberá dados do micro. Os dados antigos não são perdidos e os novos dados são gravados ao seu final, após os antigos. Não é possível ler dados, mas apenas gravá-los. A sequência de uso do arquivo nesse caso é a seguinte:

modo APPEND [
— Reabrir o arquivo;
— Acrescentar dados ao arquivo;
— Fechar o arquivo.

3 — Como entrada (INPUT) de dados.
Nesse caso o arquivo já deve existir e será aberto para enviar dados ao micro. Os dados não são perdidos e apenas a leitura é permitida. Não é possível gravar dados no arquivo. A sequência de uso do arquivo nessa situação é a seguinte:

modo INPUT [
— Abrir o arquivo;
— Ler o arquivo;
— Fechar o arquivo.

Vamos criar um arquivo e armazenar nele aquela nossa lista com três nomes e três telefones. Observe o programa da figura 6.4. Digite-o e rode-o, tendo no drive o disco que você formatou no início deste capítulo.

FIGURA 6.4 — Criação de arquivo sequencial.

```
10 OPEN"A:LISTEL.DAT" FOR OUTPUT AS #1
20 PRINT #1, "ANA 63-6205"
30 PRINT #1, "SUZY 544-6302"
40 PRINT #1, "CARLA (021)298-3202"
50 CLOSE #1
RUN
Ok
█
```

Vamos analisar rapidamente o que faz o programa da figura 6.4.

A linha 10 cria e abre (OPEN) um arquivo em disco com o nome LISTEL.DAT, avisa ao DISK BASIC que ele será usado para a gravação de dados (FOR OUTPUT) e ainda atribui a ele o número 1 (AS #1).

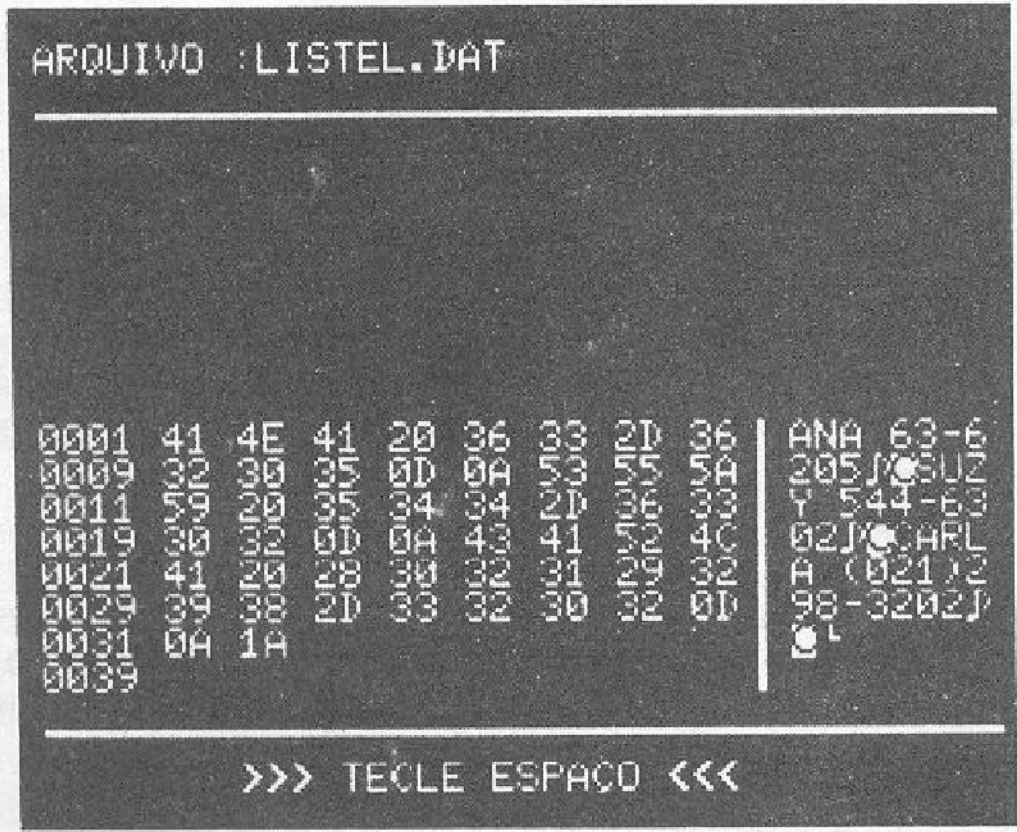
As linhas 20, 30 e 40 gravam dados no arquivo usando o comando PRINT #1. Note que o número do arquivo (#1) é fundamental.

A linha 50 apenas fecha o arquivo LISTEL.DAT.

Após a execução do programa, o arquivo "LISTEL.DAT" terá sido criado no disco e se pudéssemos ver seu conteúdo observaríamos os dados como mostrados na figura 6.5.

FIGURA 6.5 — Dados no arquivo LISTEL.DAT.

```
ARQUIVO :LISTEL.DAT
________________________________________________

0001 41 4E 41 20 36 33 2D 36 | ANA 63-6
0009 32 30 35 0D 0A 53 55 5A | 205♪●SUZ
0011 59 20 35 34 34 2D 36 33 | Y 544-63
0019 30 32 0D 0A 43 41 52 4C | 02♪●CARL
0021 41 20 28 30 32 31 29 32 | A (021)2
0029 39 38 2D 33 32 30 32 0D | 98-3202♪
0031 0A 1A                   | ●
0039
________________________________________________
          >>> TECLE ESPAÇO <<<
```

Observe que após cada dado foi gravado o separador (formado pelos caracteres CR=&H0D=13 e LF=&H0A=10) e ao final dos dados foi gravado o caractere SUB (código &H1A=26) que indica "fim do arquivo".

A única forma de se gravar dados em um arquivo sequencial é usando a instrução PRINT. Ela pode ser usada

tanto para gravar dados tipo "string" como para gravar números e, nesse caso, o uso da instrução USING é particularmente aconselhável.

Agora completaremos o nosso programa com mais algumas linhas, fazendo-o ler os dados após tê-los gravados. Digite as linhas mostradas na figura 6.6 e rode novamente o programa.

FIGURA 6.6 — Leitura de arquivo sequencial.

```
10 OPEN"A:LISTEL.DAT" FOR OUTPUT AS #1
20 PRINT #1, "ANA 63-6205"
30 PRINT #1, "SUZY 544-6302"
40 PRINT #1, "CARLA (021)298-3202"
50 CLOSE #1
RUN
60 OPEN"A:LISTEL.DAT" FOR INPUT AS #1
70 INPUT #1,A$ : PRINT A$
80 INPUT #1,B$ : PRINT B$
90 INPUT #1,C$ : PRINT C$
100 CLOSE #1
Ok
RUN
ANA 63-6205
SUZY 544-6302
CARLA (021)298-3202
Ok
█

color  auto  goto  list  run
```

Vamos mais uma vez analisar o programa, agora a partir da linha 60. Ela abre o arquivo LISTEL.DAT e avisa ao DISK BASIC que ele será usado apenas para leitura de dados.

As linhas 70, 80 e 90 lêem os dados do arquivo, os inserem nas variáveis A$, B$ e C$ e as imprime na tela.

A linha 100 simplesmente fecha o arquivo.

Para ler dados de um arquivo sequencial podemos usar o comando INPUT #1 (o #1 é indispensável!) ou ainda os comandos LINE INPUT #1 ou INPUT$(#1,n). Não vamos detalhar o funcionamento destes dois últimos comandos de leitura neste capítulo. Consulte o capítulo 8 para

aprender mais sobre eles.

Vamos agora acrescentar dados ao nosso arquivo. Para isso devemos implementar o programa com as linhas mostradas na figura 6.7. Digite-as e rode mais uma vez o programa.

FIGURA 6.7 — Abrindo arquivos para complementá-los.

```
RUN
60 OPEN"A:LISTEL.DAT" FOR INPUT AS #1
70 INPUT #1,A$ : PRINT A$
80 INPUT #1,B$ : PRINT B$
90 INPUT #1,C$ : PRINT C$
100 CLOSE #1
Ok
RUN
ANA 63-6205
SUZY 544-6302
CARLA (021)298-3202
Ok
110 OPEN"A:LISTEL.DAT" FOR APPEND AS #1
120 PRINT #1,"SHEILA 491-8766"
130 PRINT #1,"CAMILA 495-0203"
140 PRINT #1,"DALVA (031) 66-4536"
150 CLOSE #1
RUN
ANA 63-6205
SUZY 544-6302
CARLA (021)298-3202
Ok
█
color  auto  goto  list  run
```

Analisemos estas últimas linhas do programa.

A linha 110 reabre o arquivo LISTEL.DAT e avisa ao DISK BASIC que ele será usado para gravar dados. Entretanto o arquivo não deverá ser criado e os dados serão gravados no arquivo já existente a partir de seu final.

As linhas de 120 a 130 enriquecem a nossa lista telefônica com mais três dados. Após a execução dessas linhas o arquivo conterá seis dados no total.

A linha 150, finalmente, fecha o arquivo!

Agora, após tê-lo estudado detalhadamente, apague o programa que acabamos de digitar da memória do micro com um comando:

```
NEW
```

O arquivo LISTEL.DAT ainda deve estar no disco. Va-

mos fazer um outro programa para ler esse arquivo, independentemente do número de dados que ele contenha.

Digite e execute o programa listado na figura 6.8.

FIGURA 6.8 — Lendo qualquer arquivo sequencial.

```
100 SCREEN 0 : WIDTH 38 : KEY OFF
110 INPUT "QUAL O NOME DO ARQUIVO";N$
120 OPEN N$ FOR INPUT AS #1
130 IF EOF(1) = -1 THEN 180
140   LINE INPUT #1, A$
150   X = X + 1
160   PRINT X;" ==> ";A$
170   GOTO 130
180 PRINT
190 PRINT "FIM DO ARQUIVO !!!"
200 END
Ok
RUN
```

```
QUAL O NOME DO ARQUIVO? LISTEL.DAT
 1  ==> ANA 63-6205
 2  ==> SUZY 544-6302
 3  ==> CARLA (021)298-3202
 4  ==> SHEILA 491-8766
 5  ==> CAMILA 495-0203
 6  ==> DALVA (031) 66-4536

FIM DO ARQUIVO !!!
Ok
█
```

A linha 100 apenas prepara a tela como SCREEN 0 e com 38 colunas.

A linha 110 pede a introdução do nome do arquivo a ser lido. O usuário deve fornecer o nome por completo, incluindo a extensão. No nosso exemplo, será LISTEL.DAT.

A linha 120 abre o arquivo cujo conteúdo será lido.

A linha 130 faz um teste usando a função EOF do DISK BASIC MSX. Cada vez que EOF (End Of File) é usada, ela checa se o último dado do arquivo já foi lido. Em caso afirmativo, o valor de EOF é −1. Se ainda existirem dados

para serem lidos, o valor de EOF será 0. Note que é necessário especificar o número do arquivo dentro dos parênteses. Se o fim do arquivo foi encontrado, EOF(1)=1, a execução é desviada para a linha 180.

A linha 140 lê uma "linha de dados" do arquivo e a armazena na variável A$. A diferença entre o uso do INPUT e do LINE INPUT está nos "separadores" que eles reconhecem. Veja no capítulo 8 mais detalhes sobre isso.

A linha 150 apenas aumenta uma unidade na variável X. Ela, a rigor não é necessária, mas pode ser usada (como fizemos) para contar quantos registros já foram lidos. Lembre-se que ao comandar RUN todas as variáveis numéricas (como X) são inicializadas automaticamente com o valor 0. Portanto, para cada "linha de dados" lida a variável X é incrementada uma unidade, começando a contagem do 0.

A linha 160 mostra na tela o valor de X e a "linha de dados" contida em A$.

A linha 170 faz com que a execução do programa retorne à linha 130, onde a função EOF é usada para verificar se o arquivo já foi totalmente lido.

As linhas a seguir só são executadas quando o fim de arquivo (EOF) for encontrado.

A linha 180 imprime na tela uma linha em branco e a linha 190 mostra a mensagem indicando que o fim de arquivo foi encontrado.

A linha 200 contém uma instrução END. Além de terminar o programa, essa instrução fecha todos os arquivos que estiverem abertos. No nosso caso, poderíamos ter usado uma instrução CLOSE #1 ou simplesmente a instrução CLOSE sem mais nada à sua frente. A instrução CLOSE, quando usada sem a especificação do número de arquivo, fecha todos os arquivos abertos.

Isso praticamente encerra o processo de criação, abertura, gravação, atualização e leitura de arquivos sequenciais. Para compreender melhor esse poderoso recurso do DISK BASIC, consulte o capítulo 8. Mais adiante, neste mesmo capítulo, voltaremos a falar dos arquivos sequenciais. Antes, porém, vamos aprender a usar arquivos randômicos (ou aleatórios).

ARQUIVOS DE ACESSO ALEATÓRIO

Os arquivos de acesso aleatório possuem apenas um modo de abertura, porém ele é mais do que suficiente, uma vez que permite a gravação, leitura e alteração dos dados no arquivo. Lembre-se que os arquivos aleatórios têm informações de tamanhos fixos (REGISTROS), divididas em partes (CAMPOS) também de tamanhos fixos! A sequência de uso dos arquivos aleatórios é a seguinte:

— Se o arquivo não existe, criar o arquivo;
  Abrir o arquivo;
  Definir o tamanho dos registros;
— Definir os campos do arquivo;
— Gravar os campos do arquivo;
— Fechar o arquivo.

Não se assuste com a quantidade de itens, pois como veremos, o uso dos arquivos randômicos é bem fácil.

Para verificar isso, vamos criar um arquivo randômico e armazenar nele aquela nossa relação de nomes e telefones. Digite e execute o programa da figura 6.9.

FIGURA 6.9 — Criação de arquivo aleatório.

```
10 OPEN "A:LISTEL2.DAT" AS #1 LEN=20
20 FIELD #1, 20 AS A$
30 LSET A$="ANA 63-6205"
40 PUT #1,1
50 LSET A$="SUZY 544-6302"
60 PUT #1,2
70 LSET A$="CARLA (021) 298-3202"
80 PUT #1,3
90 CLOSE
```

Vamos analisar cada uma das linhas do programa.

A linha 10 verifica se o arquivo de nome LISTEL2.DAT já existe no disco. Em caso afirmativo ele apenas abre (OPEN) o arquivo para poder usá-lo. Caso o arquivo não exista, o comando OPEN o cria no disco, reservando um espaço de 20 bytes (LEN=20) para cada REGISTRO.

A linha 20 avisa ao DISK BASIC que os 20 bytes serão tratados como um só CAMPO e que a variável A$ deve

ser reservada apenas para armazenar esses dados. A variável A$ terá um tamanho fixo de 20 bytes! Note que no presente caso, o REGISTRO tem apenas um CAMPO e portanto seus tamanhos são iguais: 20 bytes.

As linhas 30, 50 e 70, atribuem dados à variável A$. O comando LSET é análogo ao LET, porém com uma pequena e fundamental diferença: LSET "alinha os dados no lado esquerdo de A$". Lembre-se que A$ tem sempre 20 bytes, nenhum a menos e nenhum a mais! Quando o dado é menor que 20 bytes, as posições vagas são preenchidas com espaços. Quando o dado é maior que 20 bytes, os excedentes são eliminados (atribuição PROCUSTIANA — veja num dicionário de mitologia grega quem foi Procustes e faça as analogias!!!). A informação pode ficar à direita ou à esquerda dos espaços. A instrução LSET faz com que ela fique obrigatoriamente à esquerda. Se quiséssemos que a informação ficasse à direita, deveríamos usar RSET no lugar de LSET.

As linhas 40, 60 e 80 gravam respectivamente os registros 1, 2 e 3 do arquivo. Note que a instrução PUT, responsável pela gravação, necessita que se especifique o número do arquivo (#1) e o número do registro (1, 2 e 3). O que é gravado é o conteúdo da variável A$. A única maneira de gravar dados num arquivo randômico é através da instrução PUT #.

O comando CLOSE da linha 90 apenas fecha o arquivo.

Após a execução do programa o arquivo terá sido criado no disco e se pudéssemos ver seu conteúdo, veríamos os dados como mostrados na figura 6.10.

FIGURA 6.10 — Dados no arquivo LISTEL2.DAT.

```
ARQUIVO  LISTEL2.DAT
------------------------------------------------
0001 41 4E 41 20 36 33 2D 36 | ANA 63-6
0009 32 30 35 20 20 20 20 20 | 205
0011 20 20 20 20 53 55 5A 59 |     SUZY
0019 20 35 34 34 2D 36 33 30 |  544-630
0021 32 20 20 20 20 20 20 20 | 2
0029 43 41 52 4C 41 20 28 30 | CARLA (0
0031 32 31 29 20 32 39 38 2D | 21) 298-
0039 33 32 30 32             | 3202
0041
```

Note que os dados estão gravados em sequência, porém de 20 em 20 bytes (tamanho do REGISTRO). Não existem separadores, como nos arquivos sequenciais.

Agora, apague o programa que está na memória com um comando:

```
NEW
```

Vamos aprender a ler arquivos randômicos.

Digite e rode o programa listado na figura 6.11.

FIGURA 6.11 — Lendo um arquivo aleatório.

```
10 OPEN "A:LISTEL2.DAT" AS #1 LEN=20
20 FIELD #1, 20 AS A$
30 GET #1,1
40 PRINT "REGISTRO 1: ";A$
50 GET #1,2
60 PRINT "REGISTRO 2: ";A$
70 GET #1,3
80 PRINT "REGISTRO 3: ";A$
90 CLOSE
```

Nesse programa, as linhas que nos interessam são as de número 30, 50 e 70. Nelas, a instrução GET # lê informações do arquivo e as insere na variável A$. A sintaxe de GET # é análoga à do PUT #. A única maneira de ler dados de arquivos aleatórios é usando GET #.

Vamos agora ver como podemos acrescentar dados no arquivo. Para isso, devemos aprender mais duas funções do DISK BASIC: LOF e LOC.

A função LOF diz o número de bytes já gravados num arquivo. Para saber o número de registros, basta dividir o número de bytes pelo tamanho do registro. Por exemplo, o arquivo LISTEL2.DAT deve ter 60 bytes, 20 para cada um dos 3 registros gravados. Obviamente, dividindo 60 (número de bytes) por 20 (tamanho do registro), obtemos 3 (número de registros).

A função LOC diz o número do último registro acessado, tanto para leitura quanto para gravação. Portanto, logo após a gravação ou leitura de um registro, a função LOC deve fornecer seu número.

Mais uma vez, apague o programa da memória co-

mandando:

NEW

Agora, digite e execute o programa da figura 6.12.

FIGURA 6.12 — Acrescentando dados em arquivos randômicos.

```
10 OPEN "LISTEL2.DAT" AS #1 LEN=20
20 FIELD #1,20 AS A$
30 PRINT "TAMANHO ANTERIOR"
40 PRINT "DO ARQUIVO:";LOF(1);"BYTES"
50 PRINT "REGISTROS JÁ"
60 PRINT "GRAVADOS:";LOF(1)/20
70 LSET A$="SHEILA 491-8766"
80 PUT #1,LOF(1)/20 + 1
90 LSET A$="CAMILA 495-0203"
100 PUT #1,LOF(1)/20 + 1
110 LSET A$="DALVA (031) 66-4536"
120 PUT #1,LOF(1)/20 + 1
130 PRINT "ÚLTIMO REGISTRO"
140 PRINT "ACESSADO:";LOC(1)
150 CLOSE
```

Desta vez não vamos analisar o programa! Faça isso você mesmo e não siga adiante enquanto não conseguir entendê-lo. Você verá que ele é bem simples!

Para gravar números em arquivos aleatórios, você pode simplesmente transformá-los em "strings" e depois gravá-los. Esse, entretanto, não é o processo mais econômico. Existem seis funções do DISK BASIC que permitem gravar e ler dados numéricos de arquivos randômicos da forma mais econômica possível: MKI$, MKS$, MKD$ e CVI, CVS e CVD. Nós não vamos detalhá-las neste capítulo mas você pode aprender um pouco mais sobre elas no capítulo 8.

USANDO VÁRIOS ARQUIVOS SIMULTANEAMENTE

Você pode usar até 6 arquivos em disco simultaneamente. Para isso, entretanto, é necessário avisar ao DISK BASIC quantos arquivos serão abertos através do comando MAXFILES. Analise o programa da figura 6.13. Ele abre dois arquivos em disco e um arquivo na impressora. Depois, os dados lidos no disco são enviados para a im-

pressora.

FIGURA 6.13 — **Usando vários arquivos simultaneamente.**

```
10 MAXFILES = 3
20 OPEN "A:LISTEL.DAT"FOR INPUT AS #1
30 OPEN "A:LISTEL2.DAT"AS #2 LEN=20
40 FIELD #2,20 AS A$
50 OPEN "LPT:"FOR OUTPUT AS #3
60 PRINT #3,"DADOS DO LISTEL.DAT:"
70 PRINT #3,"--------------------"
80 IF EOF(1)=-1 THEN 120
90      LINE INPUT #1,B$
100     PRINT #3,B$
110     GOTO 80
120 PRINT #3,""
130 PRINT #3,"DADOS DO LISTEL2.DAT:"
140 PRINT #3,"---------------------"
150 FOR F=1 TO LOF(2)/20
160     GET #2,F
170     PRINT #3,A$
180 NEXT F
190 CLOSE
```

CRIANDO ARQUIVOS BATCH COM O DISK BASIC

Uma aplicação muito útil do DISK BASIC é a criação de arquivos tipo BATCH para serem usados pelo MSXDOS. Muitas vezes, não podemos usar o comando COPY (e nem alguns editores de texto mais antigos, como o SCED e o WORDSTAR) para produzir arquivos BATCH com caracteres de controle do vídeo. Nesses casos, há duas opções: ou se usa um editor de texto mais moderno, como o MSXWRITE, ou se usa o DISK BASIC.

Por exemplo, para criar um arquivo BATCH que limpe a tela, ao invés de usarmos o comando MODE (como fizemos no capítulo 5) podemos usar o caractere de código ASCII 12 (ou &H0C). Ele não pode ser introduzido com o comando COPY. Vamos gerar um novo arquivo CLS.BAT mais rápido e mais genérico que o do capítulo 5. Digite e execute o programa da figura 6.14.

FIGURA 6.14 — Gerando o CLS.BAT com o DISK BASIC.

```
10 OPEN "A:CLS.BAT" AS #1 LEN=2
20 FIELD #1,2 AS A$
30 LSET A$=CHR$(12)+CHR$(26)
40 PUT #1,1
50 CLOSE
```

Depois de rodar o programa, transfira o arquivo CLS.BAT gerado no disco para um outro disco que possua o MSXDOS. Mais tarde, experimente esse novo CLS.

Se você quiser experimentar outros arquivos BATCH, digite e execute também os programas das figuras 6.15 e 6.16 e transfira-os para um disco com o MSXDOS.

FIGURA 6.15 — Arquivo para alterar o cursor.

```
10 OPEN "A:CURSOR2.BAT" AS #1 LEN=4
20 FIELD #1,4 AS A$
30 LSET A$=CHR$(27)+CHR$(121)+CHR$(52)
   +CHR$(26)
40 PUT #1,1
50 CLOSE
```

FIGURA 6.16 — Arquivo para restabelecer o cursor.

```
10 OPEN "A:CURSOR1.BAT" AS #1 LEN=4
20 FIELD #1,4 AS A$
30 LSET A$=CHR$(27)+CHR$(120)+CHR$(52)
   +CHR$(26)
40 PUT #1,1
50 CLOSE
```

O arquivo da figura 6.15, ao ser chamado pelo MSXDOS, faz com que o cursor apareça na tela pela metade. O arquivo da figura 6.16 faz com que o cursor volte a aparecer por inteiro.

Um outro fato importante sobre os arquivos BATCH é que eles são compatíveis com os arquivos sequenciais do DISK BASIC. Além de criar arquivos BATCH, você pode lê-los como se fossem arquivos sequenciais.

O mesmo ocorre também com os programas em BASIC gravados em ASCII através da apção ",A". Experimente

gravar um programa qualquer em BASIC com a opção ",A" e depois tente lê-lo do disco como se fosse um arquivo sequencial. Você vai ter que fazer um programa parecido com o da figura 6.8 para conseguir isso.

ALGUMAS APLICAÇÕES

Vamos apresentar a seguir alguns programas um pouco maiores que os mostrados até aqui. Entretanto, eles são bem simples para que você possa estudá-los e entendê-los. Eles estão listados no apêndice 3 com os nomes:

ESTATIST.BAS, MUSICA.BAS e LISTPRE.BAS.

Para entender seu funcionamento, rode-os operando com o disco de trabalho DESTRAVADO (janela FECHADA) para que eles possam criar nele seus arquivos de dados.

Além das instruções que vimos até agora, o DISK BASIC possui as seguintes: VARPTR, CALL VERIFY, DSKF, DSKI$ e DSKO$. Portanto, se nesses programas você se deparar com alguma instrução que não tenha sido estudada até aqui, procure esclarecer suas dúvidas consultando o capítulo 8 (você verá que muitas explicações fornecidas no dicionário referem-se a esses 3 programas).

O programa ESTATIST.BAS é um exemplo de uso de Arquivos Sequenciais, e permite gerar e armazenar tabelas de dados. Eles podem ser posteriormente recuperados (lidos) e novos dados podem ser acrescentados a uma tabela já existente. Esse programa ilustra o uso dos comandos OPEN (FOR OUTPUT, FOR INPUT e FOR APPEND), CLOSE, INPUT#, LINE INPUT# PRINT# e da função EOF.

Os programas MUSICA.BAS e LISTPRE.BAS exemplificam a utilização de Arquivos de Acesso Aleatório (Randômicos).

O primeiro gera um arquivo de músicas permitindo catalogar o nome de uma música, o conjunto que a toca, o disco em que ela está, o seu tempo de duração e o gênero. A posterior leitura poderá ser feita com critério de seleção. Esse programa ilustra o uso dos comandos OPEN (com LEN), FIELD, GET, PUT, LSET, RSET e da função LOF.

O segundo permite gerar listas de preços e ilustra a utilização das função MKI$, MKS$, MKD$, CVI, CVS e CVD.

DICIONÁRIOS DE CONSULTA

Nos capítulos 7 e 8, são apresentados os comandos do MSXDOS e do DISK BASIC.

Os comandos estão relacionados em ordem alfabética e para cada um é mostrado o modo de utilizar (sintaxe) de acordo com a seguinte convenção:

[ ] COLCHETES — As construções apresentadas entre colchetes são opcionais.

Exemplo:

```
COMANDO [ XXX ]
```

O conteúdo XXX pode ser utilizado ou não.

| BARRA VERTICAL — Indica que deve ser utilizada apenas uma das construções por ela separadas. Se usarmos uma das construções, a outra não deve ser utilizada e vice-versa.

Exemplo:

```
COMANDO XXX | YYY
```

Na utilização de COMANDO, deve-se optar ou pela construção XXX ou pela construção YYY.

{ } CHAVES — Indicam que pelo menos um dos seus conteúdos deve ser utilizado.

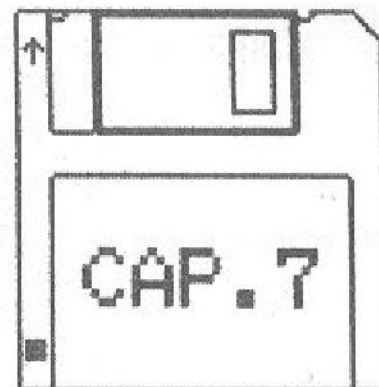
Exemplo:

```
COMANDO { XXX YYY }
```

O conteúdo XXX ou YYY (ou os dois) deve obrigatoriamente ser utilizado.

Após a apresentação do comando e de sua sintaxe, é mostrada a finalidade do mesmo e são relacionados os significados das abreviações utilizadas.

Finalmente, alguns exemplos são mostrados para elucidar a correta utilização de cada um dos comandos.

COMANDOS DO MSXDOS

BASIC [nomearq]

Permite sair do MSXDOS para entrar no DISK BASIC.

nomearq = nome de um arquivo-programa escrito necessariamente em BASIC. Se for especificado, o sistema sai do MSXDOS, vai para o BASIC e executa o programa "nomearq".

Veja o comando CALL SYSTEM para retornar do BASIC para o MSXDOS.

Exemplos:

```
A> BASIC
```

Simplesmente sai do MSXDOS e vai para o BASIC

```
A> BASIC PROGRAMA.BAS
```

Sai do MSXDOS para o BASIC e executa o programa PROGRAMA.BAS.

COPY [disp1] nomearq1 [disp2] [nomearq2] [opc]

Permite efetuar cópias de arquivos.

disp1 = dispositivo onde se encontra o arquivo a ser copiado. Pode ser qualquer drive (A:, B:, C:, D:, E: ou F:), o teclado (CON), ou o dispositivo auxiliar (AUX). O uso do próprio teclado como console permite produzir arqui-

vos "BATCH".

disp2 = dispositivo onde será feita a cópia. Pode ser qualquer drive (A:, B:, C:, D:, E: ou F:), a impressora (PRN ou LST), o vídeo (CON), o dispositivo auxiliar (AUX) ou nulo (NUL).

nomearq1 = nome do arquivo-fonte a ser copiado. Podem ser utilizados os símbolos "coringas" (* ou ?).

nomearq2 = nome do arquivo-cópia. Se for omitido, o arquivo-cópia terá o mesmo nome que o arquivo-fonte. Podem ser usados os "coringas" (* e ?).

opc = pode ser uma ou mais das seguintes opções:

/V = opção que ativa a verificação dos dados após a sua gravação.

/B = opção que avisa ao MSXDOS que todos os caracteres dos arquivos devem ser copiados. Usado para arquivos em binário.

/A = opção que avisa ao MSXDOS para não copiar o caractere correspondente a "fim de arquivo" (código 26). Usado para arquivos em ASCII.

Se houver apenas um drive físico conectado e os dispositivos especificados forem os drives A: e B: (origem e destino, respectivamente), a cópia de um disco para outro poderá ser feita mesmo assim. O MSXDOS simula a presença de dois drives lógicos e solicita a troca dos discos durante a cópia, desde que não tenha sido pressionada a tecla CONTROL enquanto o sistema estava sendo inicializado.

Exemplos:

```
A>
A>COPY CON A:EXEMPLO.BAT
REM EXEMPLO DE USO DO CONSOLE
REM (TECLADO) COMO DISPOSITIVO
REM FONTE
PAUSE Aperte uma tecla
DIR
^Z
```

Digite este exemplo pressionando RETURN no fim de

cada linha. Para gravar o arquivo "BATCH" digitado finalize pressionando CONTROL+Z (^Z) e RETURN.

```
A> COPY *.* B:/V
```

Lê todos os arquivos do drive corrente (A:) e os envia para uma cópia com verificação para o drive B:. Os arquivos-cópia terão o mesmo nome que os arquivos-fonte.

```
A> COPY B:PROGRAMA.BIN A:PROGR.BKP
```

Faz a leitura do arquivo PROGRAMA.BIN do drive B: copiando-o no drive A: com o nome PROGR.BKP.

```
A> COPY A:ARQUIVO.BAT CON
```

Copia o arquivo ARQUIVO.BAT do drive A: na tela. Note que o uso do dispositivo CON como destino corresponde ao vídeo.

Caso deseje-se enviar o arquivo ARQUIVO.BAT para a impressora deve-se comandar:

```
A> COPY A:ARQUIVO.BAT PRN
```

ou

```
A> COPY A:ARQUIVO.BAT LST
```

O comando COPY permite também que sejam emendados arquivos. Veja o capítulo 5 para maiores detalhes.

DATE [[dia-mês-ano] | [dia/mês/ano]]

Apresenta e permite alterar a data corrente.
dia = 0 a 31
mês = 1 a 12
ano = 00 a 79, 80 a 99 ou 1980 a 2099

Quando um arquivo é gravado, a data é gravada junto com ele.

Se você digitar apenas DATE e depois pressionar a tecla RETURN, será mostrada a data corrente e o sistema ficará à espera da entrada de uma nova data. Não querendo alterá-la, pressione RETURN novamente.

Ao introduzir uma data, certifique-se de que ela é válida e separe o dia, mês e ano por " — " ou por "/".

No caso do ano, valores de dois dígitos no intervalo 80 a 99 equivalem a 1980 a 1999 e valores de 00 a 79 equivalem a 2000 a 2079.

Exemplos:

```
A> DATE 20-01-1988
```

ou

```
A> DATE 20/01/88
```

DEL [drive] nomearq

Permite apagar arquivos do diretório do disco.

nomearq = nome do arquivo a ser apagado do diretório. Permite o uso dos "caracteres-coringas" (* ou ?).

Se você comandar DEL *.*, todos os arquivos do diretório serão apagados, mas antes você terá que confirmar esta opção.

Exemplos:

```
A> DEL EXEMPLO.BAS
```

Apaga do diretório o arquivo de nome EXEMPLO.BAS.

```
A> DEL *.BAT
```

Apaga do diretório todos os arquivos com a extensão BAT.

O comando DEL pode ser substituído por ERASE, valendo as mesmas regras e observações.

DIR [drive] [nomearq] [/P] [/W]

Apresenta os arquivos gravados num disco, o comprimento em bytes e a data de gravação de cada um.

Apresenta também, ao fim da listagem, o número de arquivos e o espaço livre do disco (em bytes).

drive = pode ser qualquer drive (A:, B:, C:, D:, E: ou F:). Se for omitido será assumido o drive corrente.

nomearq = nome do arquivo a ser procurado no disco. Se for omitido, serão apresentados todos os arquivos do disco. É permitida a utilização dos símbolos "coringas" (* ou ?) em "nomearq".

/P = esta opção permite a apresentação dos nomes dos arquivos na tela até que ela fique repleta. Para dar continuidade à apresentação da listagem, deve-se pressionar uma tecla qualquer.

/W = esta opção faz com que sejam apresentados apenas os nomes dos arquivos. Os comprimentos (em bytes) e as datas de gravação não são mostrados.

Os caracteres "coringas" (* ou ?) podem ser usados no nome do arquivo para selecionar os que tem características comuns. Para facilitar o uso deles vale a seguinte correspondência

DIR	equivale a	DIR *.*
DIR nomearq	equivale a	DIR nomearq.*
DIR .EXT	equivale a	DIR *.EXT
DIR .	equivale a	DIR *.

Exemplos:

```
A> DIR*.COM/W
```

ou

```
A> DIR .COM/W
```

Lista todos os nomes dos arquivos com extensão COM. Os comprimentos e datas de gravação não aparecem devido à opção /W estar ativada. O drive é o corrente (A:).

```
A>DIR B:C*.* /P
```

Lista no drive B: todos os arquivos que comecem com a letra C. Espera a pressão de uma tecla quando a tela ficar cheia (opção /P ativada).

O comando DIR aceita o complemento ^P (CONTROL + P) e ^N (CONTROL + N) para ativar e desativar, respectivamente, o "eco" na impressora.

ERASE [drive] nomearq

Veja o comando DEL.

FORMAT

Permite que seja efetuada a formatação de um disco, para que ele aceite arquivos do MSXDOS.

Todo disco virgem deve ser formatado antes que possa ser utilizado pelo MSXDOS ou pelo DISK BASIC.

Ao utilizar o comando FORMAT o disco terá o diretório e a tabela de alocação de arquivos (FAT — File Allocation Table) inicializados. Além disso, após a formatação, não haverá qualquer arquivo no disco. Portanto, certifique-se de que não há informações importantes gravadas num disco antes de formatá-lo, pois elas serão irremediavelmente perdidas após esse procedimento.

Quando o comando FORMAT é executado, deve ser escolhido o drive onde se encontra o disco a ser formatado (A, B, C ou D) e o tipo de formatação a ser efetuada (40 ou 80 trilhas com face simples ou dupla).

No caso de querer-se formatar um disco de 5 1/4" escolha formatação com 40 trilhas. Se o disco for de 3 1/2", a opção deve ser por 80 trilhas.

Exemplo:

```
A>FORMAT
```

MODE numcol

Permite definir o número máximo de colunas que poderão existir na tela.

numcol = número máximo de colunas (caracteres por linha).

O número máximo de colunas pode variar entre 1 e 40 nos microcomputadores MSX modelo 1 sem placa de 80 colunas. No modelo 2 ou modelo 1 com placa de 80 colunas, esse número pode estar entre 32 e 80.

Exemplo:

```
A> MODE 39
```

PAUSE [mensagem]

Permite a interrupção da execução de um arquivo do tipo "BATCH".

Durante a execução de um arquivo "BATCH", você pode ter a necessidade de trocar discos, ligar a impressora ou qualquer outra ação. O comando PAUSE suspende a execução do "BATCH" e a mensagem (se especificada) é mostrada na tela até que uma tecla qualquer seja pressionada.

Exemplo:

```
A> PAUSE Coloque Disco no Drive A
```

REM comentários

Permite apresentar mensagens na tela durante a execução de um arquivo "BATCH" sem interompê-la.

Exemplo: veja o primeiro exemplo do comando COPY.

REN [drive] nomearq1 nomearq2
RENAME [drive] nomear1 nomearq2

Permite alterar o nome de um arquivo gravado num disco.

drive = A:, B:, C:, D:, E: ou F:. Se não for especificado será assumido o drive corrente.
nomearq1 = nome do arquivo a ser trocado.
nomearq2 = novo nome a ser dado para o arquivo.

Os caracteres "coringas" (* ou ?) podem ser utilizados tanto em nomearq1 como em nomearq2. Quando usados em nomearq2 (novo nome), os caracteres correspondentes às posições em que foram colocados coringas não

são alterados.

O comando REN não aceita como novo nome algum que já exista gravado no disco.

Exemplos:

```
A> REN EXEMPLO.BAS PROGRAM.BAS
```

Troca o nome do arquivo EXEMPLO.BAS para PROGRAM.BAS.

```
A> RENAME *.GAM *.BIN
```

Troca todas as extensões GAM por BIN, sem alterar os nomes dos arquivos.

```
A> REN B:PISKAZOR.BAS L??T?D??.*
```

Altera as letras P, K e Z para L, T e D, respectivamente. Assim o arquivo PISKAZOR.BAS presente no drive B: será renomeado para LISTADOR.BAS.

TIME [h[= min[= s]]]

Apresenta e permite alterar o instante indicado pelo relógio interno do micro.

h = hora (0 a 23 ou 1 a 12 seguido de A ou P)
min = minuto (0 a 59)
s = segundo (0 a 59)

O comando TIME só funciona em micros que possuam relógio interno.

Exemplo:

```
A>TIME 16:45
```

TYPE [drive] nomearq

Apresenta na tela o conteúdo de um arquivo.

drive = A:, B:, C: ou D:. Se omitido, será assumido o drive corrente.
nomearq = nome do arquivo a ser examinado. Deve estar gravado no formato ASCII.

Para interromper ou retomar a apresentação do arquivo na tela deve-se pressionar CONTROL+S (^S).

A pressão de CONTROL+C (^C) interrompe definitivamente a apresentação.

Caso deseje-se enviar a listagem do arquivo também para a impressora, deve-se pressionar CONTROL+P (^P).

Exemplo

```
A> TYPE EXEMPLO.BAT
```

VERIFY { ON | OFF }

Liga ou desliga a verificação dos dados na gravação.

VERIFY ON ativa a verificação. Desta forma, os dados serão lidos e verificados após serem gravados no disco.

VERIFY OFF desativa a verificação.

Exemplo:

```
A> VERIFY ON
A> COPY PROGRAM.BAS B:
A> VERIFY OFF
```

Estando ativada a verificação, torna-se redundante usar a opção "/V" após o comando COPY.

COMANDOS DO DISK BASIC

BLOAD "{[disp] [nomearq]}" [, R|, S|, desloc]

Permite carregar um arquivo binário gravado em fita cassete ou em disco.

disp = nome do dispositivo onde está gravado o arquivo a ser carregado. Deve ser CAS: para cassete e A:, B:, C:, D:, E: ou F: para os drives. Se for omitido, será assumido o drive corrente.

nomearq = nome do arquivo a ser carregado. Ele deve ter sido gravado necessariamente com a instrução BSAVE. A sua especificação é obrigatória no caso do uso de um drive. Pode ser omitido quando o carregamento é feito a partir do gravador.

,R = opção que faz o arquivo (programa) ser automaticamente executado assim que ele for carregado na memória do micro.

,S = opção que permite o carregamento na VRAM de uma imagem de vídeo salva por BSAVE.

desloc = deslocamento. Se for omitido, o arquivo é carregado na memória a partir do endereço que havia sido especificado no BSAVE por ocasião da gravação. Se o deslocamento for especificado, o arquivo é carregado a partir do endereço fornecido pelo BSAVE, somado ao deslocamento. Note que, em se tratando

de um programa em Linguagem de Máquina, ele deve ser relocável para que o desloc possa ser usado sem problemas.

Exemplos:

```
BLOAD "TELA.SCR",S
```

Carrega, do drive corrente, o arquivo TELA.SCR diretamente na VRAM.

```
BLOAD "B:JOGO.BIN",R
```

Carrega, a partir do drive B, o programa JOGO.BIN e o executa.

```
BLOAD "PROGR.BIN",1024
```

Carrega o arquivo PROGR.BIN a partir do endereço especificado por BSAVE, somado com 1024.

Veja também BSAVE.

BSAVE "[disp] nomearq", endin, endfin [, endexecl, S]]

Permite gravar um arquivo binário em fita cassete ou em disco.

disp = nome do dispositivo onde será gravado o arquivo binário. Deve ser CAS: para cassete e A:, B:, C:, D:, E: ou F: para os drives. Se for omitido, é assumido o drive corrente.

nomearq = nome do arquivo binário (programa em LM, dados, tela, etc) a ser gravado.

endin = endereço inicial. Representa o endereço do primeiro byte da área da memória a ser gravada.

endfin = endereço final. Representa o endereço do último byte da área da memória ser gravada.

endexec = endereço de execução. É o endereço do byte onde começará a execução do programa. Se endexec for omitido será assumido endin como endereço de execução.

,S = esta opção permite a gravação de uma área da VRAM (memória de vídeo).

Exemplos:

```
BSAVE "TELA1.SCR",0,&H3FFF,S
```

Grava, no drive corrente, a região da VRAM que vai do endereço 0 ao 3FFF (em hexadecimal) com o nome TELA1.SCR.

```
BSAVE "B:FILTRO.BIN",37000,&H38000,37500
```

Grava um programa em Linguagem de Máquina localizado entre os endereços 37000 e 38000 e cujo endereço de execução é 37500.

CALL FORMAT ou _FORMAT

Permite formatar um disco para que possa ser utilizado pelo DISK BASIC ou pelo MSXDOS.

Todos os discos virgens devem ser formatados antes de serem usados. A presença de um disco não formatado no drive simplesmente não é detectada pelo sistema.

Cuidado ao formatar seus discos! Se for usado um disco gravado ao invés de um virgem, todas as informações nele contidas serão destruidas!

Exemplo:

```
CALL FORMAT
```

Apresenta um menú de opções e formata o disco. Lembre-se que as opções de 40 trilhas se referem a discos de 5 1/4" e as de 80 trilhas aos de 3 1/2". Os acionadores de face dupla permitem ambas as opções (1 ou 2 faces).

CALL SYSTEM ou _SYSTEM

Permite sair do DISK BASIC e voltar para o MSXDOS.

Para que este comando funcione é necessário que o MSXDOS já tenha sido carregado e que o DISK BASIC tenha sido chamado a partir dele.

O disco contendo o MSXDOS deve estar no drive para que a volta para ele seja possível.

Com a execução deste comando todos os arquivos

de dados são fechados e todos os programas e dados contidos na RAM são apagados.

Exemplo:

```
_SYSTEM
```

CALL VERIFY {ON|OFF} ou _VERIFY {ON|OFF}

Liga ou desliga a verificação dos dados na gravação.

CALL VERIFY ON ativa a verificação por ocasião da cópia de arquivos. CALL VERIFY OFF desativa esta verificação.

Exemplo:

```
10 CALL VERIFY ON
20 COPY "A:*.BAS" TO B:*.BKP
30 CALL VERIFY OFF
```

Grava todos os arquivos do drive A com a extensão BAS para o drive B com o mesmo nome e a terminação BKP. Verifica se a gravação foi fiel e desativa a rotina de verificação.

CLOSE [# numarq1] [, # numar2] [,...]

Permite fechar arquivos de dados em uso.

numarq = número do arquivo a ser fechado (1 a 15). Se for omitido, serão fechados todos os arquivos abertos pela instrução OPEN.

Os comandos END, CLEAR, NEW, MAXFILES e RUN executam o fechamento automático dos arquivos, dispensando o CLOSE.

Exemplo:

```
CLOSE #1,#2
```

Fecha os arquivos 1 e 2.

COPY "[drive1] nomearq1" TO "{[drive2] [nomearq2]}"

Permite efetuar cópia de arquivos de um disco para

outro.

drive1 = drive lógico onde está o arquivo a ser copiado (A:, B:, C:, D:, E: ou F:). Se for omitido, será assumido o drive corrente.

nomearq1 = nome do arquivo a ser copiado. Pode-se usar os coringas "*" e "?".

drive2 = drive lógico em que será feita a cópia. Se for omtido, será considerado igual ao drive1.

nomearq2 = nome do arquivo-cópia a ser gerado. Se for omitido será considerado igual ao nomearq1. Pode-se usar "*" e "?".

Quando houver apenas um drive físico conectado à interface, pode-se utilizar o comando COPY como se existissem dois drives. Neste caso as letras representarão os dois discos e as mensagens na tela do micro solicitarão a troca sempre que necessário.

Exemplos:

```
COPY "A:J*.*" TO "B:"
```

Copia do drive (disco) A todos os arquivos cujo nome comece com a letra J e os transfere para o drive (disco) B mantendo o mesmo nome e extensão.

```
COPY "A:*.BAS" TO "A:*.BKP"
```

Gera, no mesmo disco, uma cópia BACKUP de todos os arquivos cujo nome tenha a extensão BAS. As cópias terão o mesmo nome, porém com a extensão BKP.

CVI, CVS, CVD

Convertem, respectivamente, uma variável string de 2, 4 e 8 bytes numa variável numérica inteira, de precisão simples e de precisão dupla.

Normalmente as sintaxes dessas funções são:

varnum inteira = CVI (varstr de 2 bytes)
varnum de precisão simples = CVS (varstr de 4 bytes)
varnum de precisão dupla = CVD (varstr de 8 bytes)

varnum = variável numérica
varstr = variável string (alfanumérica)

Num arquivo de acesso aleatório os valores numéricos são gravados na forma de variáveis string (veja MKI$, MKS$ e MKD$). Quando forem lidos, devem novamente ser convertidos para a forma numérica por CVI (para números inteiro), CVS (para os de precisão simples) e CVD (para os de precisão dupla).

Exemplo:

Veja as linhas 510, 520 e 530 do programa LISTAPRE.BAS listado no apêndice 3.

DSKF (número do drive)

Função que permite a obtenção do número de blocos lógicos livres de um disquete.

O número do drive deve ser:

0 para o drive corrente (default)
1 para o drive A
2 para o drive B
3 para o drive C
4 para o drive D
5 para o drive E
6 para o drive F

Exemplo:

```
PRINT "BLOCOS LIVRES=";DSKF(0)
```

Mostra, na tela do micro, o número de blocos livres no drive corrente. Cada bloco pode ser formado por 1 ou 2 setores do disco, dependendo do tipo de formatação.

DSKI$ (número do drive)

Função que permite a leitura de um setor do disco.

A sintaxe dessa função é:

variável string = DSKI$ (número do drive, número do setor)

O número do drive deve estar entre 0 e 6 (veja DSKF) e o número do setor varia, para os discos de 3 1/2", entre

0 e 719 para formatação em face simples e entre 0 e 1439 para face dupla.

Os 512 bytes lidos correspondentes ao conteúdo de um setor do disco, são armazenados numa área da memória RAM do micro cujo primeiro endereço pode ser calculado pela expressão:

```
PEEK(&HF351)+256*PEEK(&HF352)
```

Exemplo:

```
100 KEYOFF:CLS
110 A$=DSKI$(0,0):E=PEEK(&HF351)+
    256*PEEK(&HF352)
120 NS=256*PEEK(E+20)+PEEK(E+19)-1
130 PRINT"Qual setor ( 0 a";NS;
140 INPUT") ";S:PRINT
150 IF S<0 OR S>NS THEN BEEP:GOTO 130
160 CLS:PRINT"SETOR";S
170 A$=DSKI$(0,S)
180 EN=PEEK(&HF351)+256*PEEK(&HF352)
190 FOR F=1 TO 64
200   Y$="":Z$=""
210   FOR G=0 TO 7
220     X=PEEK(EN+8*(F-1)+G)
230     Y$=Y$+" "+RIGHT$("0"+HEX$(X),2)
240     IF X>31 THEN Z$=Z$+CHR$(X) ELSE
             Z$=Z$+CHR$(1)+CHR$(64+X)
250   NEXT G
260   PRINT Y$;"  ";Z$;
270   IF F/22=F\22 THEN H$=INPUT$(1)
280   PRINT
290 NEXT F
300 H$=INPUT$(1):CLS:GOTO 130
```

DSKO$ número do drive, número do setor

Função que permite efetuar a gravação direta de dados em um setor do disco.

O número do drive e do setor são os especificados nos verbetes DSKF e DSKI$.

Muito cuidado deve ser tomado no uso desta função pois o DSKO$ pode alterar dados úteis gravados ou até inutilizar totalmente o disco do ponto de vista lógico.

Os bytes da RAM (até 512) a serem gravados devem

ter seu primeiro endereço apontado por (veja DSKI$):

&HF351 e &HF352

Exemplo:

```
100 KEYOFF:CLS:CLEAR 600
110 INPUT "QUAL O SEU NOME ";N$
120 N$=" * ESTE DISQUETE PERTENCE A "+
    N$
130 A$=DSKI$(0,0): E=PEEK(&HF351)+256*
    PEEK(&HF352)
140 US=256*PEEK(E+20)+PEEK(E+19)-1
150 X=LEN(N$)
160 FOR I=1 TO 512 STEP X
170 FOR J=1 TO X
180 EN=60307! +I+J: IF EN>60820! THEN
    210
190 POKE EN,ASC(MID$(N$,J,1))
200 NEXT J : NEXT I
210 MSB=INT(60309!/256)
220 LSB=60309!-256*MSB
230 POKE &HF351,LSB
240 POKE &HF352,MSB
250 DSKO$ 0,US
Ok
█
```

Este programa grava seu nome no último setor do disco.

EOF (número do arquivo)

Função que indica se o final de um arquivo sequencial aberto para leitura (FOR INPUT) foi encontrado.

Quando o final de um arquivo sequencial é encontrado, a função EOF (Enf Of File) retorna o valor −1. Caso contrário retorna o valor 0.

Essa função deve ser utilizada apenas para arquivos de acesso sequencial abertos como entrada (FOR INPUT) e deve ser colocada no programa antes de cada comando INPUT # para se evitar o erro "Input past end".

Exemplo:

Veja a linha 1340 do programa ESTATIST.BAS listado no apêndice 3.

FIELD [#] numarq, numbytes AS va1$ [, numbytes AS var2$] [,..]

Reserva uma área do buffer dos arquivos de acesso aleatório para variáveis string.

numarq = número do arquivo. Deve ser o mesmo que foi declarado pelo comando OPEN para abertura do arquivo.

numbytes = tamanho, em bytes, a ser reservado para cada variável string. A soma dos "numbytes" não deve exceder o valor declarado pelo LEN no comando OPEN.

varn$ = nome da variável necessariamente do tipo string. Esse nome deve ser de uso exclusivo do arquivo e não pode ser usado para outros fins no programa. Consequentemente esse nome não pode ser associado a comandos LET ou INPUT.

O uso do comando FIELD é obrigatório antes da leitura (GET) ou gravação (PUT) de dados num arquivo de acesso aleatório.

Exemplo:

Veja a linha 140 do programa LISTAPRE.BAS e a linha 1050 do programa MUSICA.BAS, listados no ap. 3.

FILES ["[drive] [nomearq]"]

Comando que permite a apresentação, na tela, dos nomes dos arquivos de um disco.

drive = A:, B:, C:, D:, E: ou F:. Se for omitido, será assumido o drive corrente.

nomearq = nome do arquivo. Podem ser usados os caracteres "*" e "?". Se for omitido, serão mostrados todos os nomes dos arquivos gravados no disco.

Para listar essa relação de nomes na impressora, use o comando LFILES com a mesma sintaxe e observações do FILES.

Exemplos:

FILES

Lista todos os nomes dos arquivos do drive corrente.

FILES "B:*.BIN"

Lista todos os nomes dos arquivos com extensão BIN do disco presente no drive B.

LFILES "C*.*"

Lista, na impressora, todos os arquivos cujo nome comece pela letra C, contidos no disco do drive corrente.

FORMAT

Veja CALL FORMAT.

GET [#] numarq [, numreg]

Permite efetuar a leitura de um registro de um arquivo de acesso aleatório aberto.

numarq = número do arquivo de acesso aleatório igual ao declarado pelo comando OPEN.

numreg = número do registro a ser lido. Se for omitido, será considerado o sucessivo ao último registro lido ou gravado anteriormente.

Após a execução do comando GET, os dados são armazenados numa área da RAM reservada como "buffer" de arquivos. Para obtê-los, utilize as variáveis strings declaradas pelo comando FIELD.

Exemplo:

Analise as linhas 1040, 1050, 1790, 1850, 1890, do programa MUSICA.BAS e também as linhas 140, 150, 500, 510, 520 e 530 do programa LISTAPRE.BAS, ambos listados no apêndice 3.

Se for usá-los, utilize um disco de trabalho destravado para que eles possam criar seus arquivos nele.

Opere no DISK BASIC, ou seja, ao ligar o micro não pressione a tecla SHIFT.

INPUT # numarq, var1 [, var2] [,..]

Permite que se efetue a leitura dos dados de um arquivo de acesso sequencial aberto por OPEN (FOR INPUT).

numarq = número do arquivo sequencial a ser lido. Deve ser o mesmo número declarado no comando OPEN.

varn = variável numérica ou string à qual se associa o dado a ser lido.

Os blocos de dados devem aparecer exatamente como se fossem introduzidos pelo comando INPUT.

Os caracteres:

CHR$(&H0D) = CR (Carriage Return)
CHR$(&H0A) = LF (Line Feed)
CHR$(&H20) = SP (SPace = Espaço em branco)

são ignorados durante a leitura. O primeiro caractere encontrado diferente de um dos mencionados será considerado como início dos dados.

O fim de um dado é definido por um dos 3 caracteres citados ou pela vírgula. Isso também ocorre com o espaço e o ponto-e-vírgula no caso de um dado numérico. Se um dado for do tipo string, seu fim é também determinado pela leitura de 255 caracteres.

Se os caracteres forem colocados entre aspas ("") serão lidos como um único dado, independentemente da eventual existência de vírgula entre eles.

Se, durante a leitura, for encontrado um caractere:

CHR$(&H1A) = SUB (SUBstitute character)

indicativo de fim-de-arquivo, a execução do comando:

INPUT#

é interrompida.

Exemplo:

Veja a linha 1350 do programa ESTATIST.BAS listado no apêndice 3.

INPUT$ (numcar [, [#] numarq])

Função que permite efetuar a leitura de um certo número de caracteres de um arquivo de acesso sequencial

aberto pelo comando OPEN para entrada (FOR INPUT).

numcar = número de caracteres a serem lidos.

numarq = número do arquivo de acesso sequencial aberto para entrada pelo comando OPEN. Se for omitido, os caracteres serão lidos a partir do teclado.

Os caracteres lidos devem ser associados a uma variável string.

Exemplo:

```
A$ = INPUT$(3,#1)
```

A variável A$ conterá os 3 próximos caracteres a serem lidos do arquivo 1 que deve ter sido previamente aberto por OPEN para entrada.

KILL "[drive] nomearq"

Comando que apaga arquivos do diretório de um disco.

drive = A:, B:, C:, D:, E: ou F:. Se for omitido será assumido o drive corrente.

nomearq = nome do arquivo a ser eliminado do diretório. Podem ser utilizados o "*" e "?".

Um arquivo aberto pela sentença OPEN não pode ser apagado.

Exemplo:

```
KILL "*.*"
```

Apaga todos os arquivos do diretório do disco colocado no drive corrente sem pedir confirmação (cuidado!)

LFILES

Veja FILES.

LINE INPUT # numarq, var$

Comando que permite a leitura de uma série de caracteres (até 254) de um arquivo de acesso sequencial associando-a a uma variável string.

numarq = número do arquivo de acesso sequencial aberto por OPEN.

var$ = nome da variável string à qual será associada a sequência de caracteres lida pelo comando.

Este comando lê todos os caracteres do arquivo até um máximo de 254 ou até que seja encontrada a sequência de caracteres:

CHR$(&H0D) = CR (Carriage Return)
CHR$(&H0A) = LF (Line Feed)

e os associa à variável var$.

Os caracteres CR e LF apenas funcionam como delimitadores e não são incluídos na variável var$.

Pode-se utilizar esse comando para ler as linhas de um arquivo-programa salvo em formato ASCII pelo comando SAVE.

Exemplo:

Veja a linha 1320 do programa ESTATIST.BAS no apêndice 3.

LOAD "{[disp] [nomearq]}" [,R]

Permite carregar um programa em BASIC de uma fita cassete ou de um disco.

disp = nome do dispositivo a partir do qual queremos carregar o programa. Deve ser CAS: para o gravador cassete e A:, B:, C:, D:, E: ou F: para os drives. Se for omitido, é assumido o drive corrente. Se o dispositivo for CAS:, o programa BASIC deve estar gravado em formato ASCII.

nomarq = nome do programa a ser carregado. Ele deve ter sido gravado, necessariamente, pelo comando SAVE. Sua especificação é obrigatória usando-se drive e dispensável para cassete.

,R = opção que faz o programa ser automaticamente executado assim que termina o processo de leitura.

Arquivos que não correspondam a programas em BASIC MSX mas que tenham sido gravados em formato

ASCII, podem ser carregados na memória do micro pelo comando LOAD mas não podem ser executados.
Exemplos:

```
LOAD "PROGRAMA.BAS"
```

Carrega, do drive corrente, o programa PROGRAMA. BAS.

```
LOAD "B:XYZ.ASC",R
```

Carrega, do drive B, o programa XYZ.ASC e o executa a seguir.

LOC (numarq)

Função que fornece o número do último registro gravado ou lido de um arquivo de acesso aleatório.
numarq = número do arquivo aberto.
Exemplo:
Veja a linha 610 do programa LISTAPRE.BAS gravado no disco mestre.

LOF (numarq)

Função que indica qual o comprimento, em bytes, de um arquivo.
numarq = número do arquivo aberto.
Exemplo:
Veja a linha 300 do programa LISTAPRE.BAS e a linha 1260 do programa MUSICA.BAS listados no ap. 3.

LSET var$ = expressão.string
RSET var$ = expressão string

Comando que move os dados contidos numa expressão string para a variável declarada pelo comando FIELD (buffer de um arquivo de acesso aleatório). Os dados movidos são alinhados à esquerda pelo LSET e à direita pelo RSET.

var$ = nome da variável string declarada pelo comando FIELD.

O comando LSET alinha os dados à esquerda e o comando RSET os alinha à direita. Se eles forem em menor número do que aquele que o comando FIELD reservou para var$, as posições vagas são completadas com espaços em branco. Se forem em maior número, os dados excedentes são simplesmente ignorados.

Dados numéricos devem ser convertidos para strings antes do uso de LSET ou RSET (veja MKI$, MKS$ e MKD$).

O uso de LSET ou RSET é fundamental para a preparação dos dados antes de gravá-los num arquivo de acesso aleatório com o comando PUT.

Exemplo:

Veja as linhas 1430, 1440 e 1450 do programa MUSICA.BAS e as linhas 380 a 410 do programa LISTAPRE.BAS.

MAXFILES = expressão numérica

Comando que especifica o número máximo de arquivos que podem ser abertos simultaneamente.

A expressão numérica deve resultar numa variável ou constante numérica entre 0 e 15.

Se MAXFILES=0 for comandado, somente SAVE e LOAD poderão ser executados.

O número máximo de arquivos em disco que podem ser abertos ao mesmo tempo é 6, mesmo que o valor especificado por MAXFILES seja maior que 6.

Exemplo:

```
MAXFILES = 3
```

MERGE "{[disp] [nomearq]}"

Permite juntar um programa em BASIC gravado em formato ASCII com outro existente na memória do micro.

disp = CAS: para cassete ou A:, B:, C:, D:, E: ou F: para os drives.

nomearq = nome do programa em BASIC gravado em ASCII a ser sobreposto àquele armazenado na RAM. Não pode ser omitido se o dispositivo for um drive.

Se houver coincidência no número de linhas entre o programa gravado e o existente na memória, prevalecem as do primeiro.

Exemplo:

```
MERGE "SUBROT.BAS"
```

Veja também o comando SAVE.

MKI$, MKS$, MKD$

Convertem, respectivamente, uma variável numérica inteira, de precisão simples e de precisão dupla em uma variável string de 2, 4 e 8 bytes.

Normalmente as sintaxes dessas funções são

varstr = MKI$ (variável númerica inteira)
varstr = MKS$(variável numérica de precisão simples)
varstr = MKD$(variável numérica de precisão dupla)

varstr é uma variável string qualquer.

Todos os valores numéricos a serem gravados num arquivo de acesso aleatório devem ser previamente convertidos para variáveis strings. Veja CVI, CVS e CVD para saber como recuperar (ler) esses valores. Veja também as funções LSET e RSET.

As variáveis string utilizadas na atribuição das funções MKI$, MKS$ e MKD$ conterão respectivamente 2, 4 e 8 bytes.

Exemplo: Veja as linhas 390, 400 e 410 do programa LISTPRE.BAS listado no apêndice 3.

NAME "[drive] nomearq1" AS "[drive] nomearq2"

Permite alterar o nome de um arquivo.

drive = drive onde está o arquivo a se renomeado (A:, B:, C:, D:, E: ou F:). Deve ser o mesmo nas duas opções. Se não for especificado, será assumido o drive corrente.

nomearq1 = nome a ser alterado do arquivo gravado no disco.
nomearq2 = novo nome a ser atribuído ao arquivo nomearq1.

Os caracteres "*" e "?" podem ser usados em nomearq1 e nomearq2.

Exemplos:

```
NAME "VELHO.BAS" AS "NOVO.BAS"
NAME "B:*.BAK" AS "B:*.BKP"
```

OPEN "{[disp] [nomearq]}" [FOR modo] AS [#] numarq [LEN = n]

Comando que efetua a abertura de um arquivo para permitir operações de entrada e saída de dados.

disp = LPT: (impressora), GRP: (tela gráfica), CRT: (tela de texto), CAS: (cassete), A:, B:, B:, D:, E: ou F: (drives). Se for omitido, será assumido o drive corrente.

nomearq = nome do arquivo. Só pode ser omitido para os dispositivos LPT:, GRP:, CRT: e CAS: no modo OUTPUT (saída).

modo =
- OUTPUT para arquivo sequencial como saída (gravação) que ainda não existe.
- INPUT para arquivo sequencial como entrada (leitura).
- APPEND para arquivo sequencial como saída (gravação) que já existe. Nesse modo, os dados são gravados após o último dado existente no arquivo gravado no disco.

Se modo for omitido, o arquivo aberto será de acesso aleatório para operações de saída ou entrada. Nesse caso deve-se especificar o tamanho n do registro (LEN=n). O comando FIELD também deverá ser utilizado nesse caso.

numarq = número do arquivo. Não deve exceder aquele especificado por MAXFILES.
n = tamanho do registro em bytes. A opção LEN=n só pode ser usada em arquivos randômicos.

Exemplos:

Veja as linhas 1310, 1460 e 1550 do programa ESTATIST.BAS para arquivo sequencial. Para arquivos randômicos, veja a linha 1040 do programa MUSICA.BAS e a linha 140 do programa LISTPRE.BAS, listados no ap. 3.

PRINT # numarq, exp1], exp2[[,...]

Permite que se efetue a gravação dos dados de um arquivo de acesso sequencial aberto por OPEN como saída (FOR OUTPUT ou FOR APPEND).

numarq = número do arquivo no qual serão gravados os dados. Deve ser o número declarado pelo comando OPEN.
expn = expressão (variável ou constante) numérica ou string a ser gravada como dado.

A utilização de USING também é permitida (PRINT # USING) com a mesma sintaxe do comando PRINT.

O comando PRINT # grava no disco uma imagem dos dados similar à que seria mostrada na tela com o comando PRINT. Por isso, alguns cuidados devem ser tomados para delimitar os dados no disco, de modo que posteriormente eles possam ser recuperados pelos comandos INPUT # ou LINE INPUT # sem erros.

As expressões numéricas ou strings são sempre separadas se forem colocadas isoladamente no comando PRINT #.

Exemplo:

```
A=1:B=2:PRINT#1,A:PRINT#1,B
A$="α":B$="β":PRINT#1,A$:PRINT#1,B$
```

Para as expressões numéricas, o ponto-e-vírgula (;) separa os dados:

Observação: Se vírgulas forem usadas no lugar do ponto-e-vírgula, os dados numéricos também serão separados, porém os espaços em branco que são inseridos entre os campos de impressão também serão gravados.

As expressões strings entre vírgulas ou ponto-e-vírgulas não são separadas. A separação deve ser feita por vírgulas colocadas entre aspas.

Exemplo:

```
    PRINT#1,A$; : PRINT#1,B$  ┐
ou                            ├ gera  αβ
    PRINT#1,A$;B$        gera ┘
```

```
    PRINT#1,A$, : PRINT#1,B$  ┐
ou                            ├ gera  α       β
    PRINT#1,A$,B$             ┘       └───────┘
                                      7 espaços
```

```
    PRINT#1,A$;",";B$         ] gera  α
                                      β
```

Se uma expressão string tiver em seu interior **vírgulas**, ponto-e-vírgulas, espaços em branco, códigos CR (&H0D = 13) ou LF (&H0A = 10), ela deve ser colocada entre aspas (código &H22 = 34) para ser gravada e lida por INPUT # como um único bloco de dados. Esse cuidado é desnecessário se a leitura for feita por LINE INPUT #.

Por exemplo:

```
A$="EDITORA ALEPH , TECHNOAHEAD"
```

deve ser ser gravada como:

```
PRINT#1,CHR$(34);A$;CHR$(34)
```

PUT [#] numarq, [numreg]

Permite efetuar a gravação de um registro de um arquivo de acesso aleatório aberto.

numarq = número do arquivo. É o número segundo o qual o arquivo foi aberto pela instrução OPEN.

numreg = número do registro a ser gravado. Se for omitido, será considerado o registro seguinte àquele gravado (ou lido) anteriormente.

Antecedendo a execução do comando PUT, os dados devem ser movidos para uma área da RAM reservada como "buffer" de arquivos. Para isso, utilize os comandos LSET ou RSET associados às variáveis strings declaradas pelo comando FIELD.

Exemplo: Veja as linhas 1040, 1050, 1430, 1440, 1450 e 1460 do programa MUSICA.BAS. Veja também as linhas 140, 150, 380, 390, 400, 410 e 420 do programa LISTAPRE.BAS listados no apêndice 3.

RSET

Veja LSET.

RUN ["[drive] nomearq" [,R]] | [numlin]

Comando que permite a execução de um programa em BASIC.

Se nenhuma das opções for especificada, o comando RUN executa o programa em BASIC que estiver na memória do micro a partir de sua primeira linha.

drive = nome do drive a partir do qual será carregado o arquivo. Se for omitido, será assumido o drive corrente.

nomearq = nome do arquivo que deverá ser carregado do disco e automaticamente executado. Esse arquivo deve ser necessariamente um programa BASIC.

numlin = número da linha a partir da qual será executado o programa.

O comando RUN fecha todos os arquivos abertos (veja CLOSE) se a opção ",R" não for especificada.

Exemplos:

```
RUN "MATFIN.BAS"
```

Carrega do drive B: e depois executa o programa MATFIN.BAS.

```
RUN "DISCO.BAS"
```

Carrega e executa o programa DISCO.BAS.

SAVE "[disp] nomearq" [,A]

Permite gravar um programa em BASIC da memória do micro para fita cassete ou disco.

disp = nome do dispositivo onde será gravado o programa em BASIC. Deve ser CAS: para cassete e A:, B:, C:, D:, E: ou F: para drives. Se for omitido será assumido o drive corrente. Se for CAS:, o programa será gravado em ASCII, dispensando a opção ",A".

nomearq = nome do programa em BASIC a ser gravado.

,A = opção que faz com que a gravação em disco seja efetuada no formato ASCII.

Exemplos:

```
SAVE "XYZ.BAS"
```

Grava o programa XYZ.BAS no drive corrente em formato binário compactado.

```
SAVE "B:XYZ.BAS",A
```

Grava o programa XYZ.BAS no drive B: em formato ASCII.

SYSTEM

Veja CALL SYSTEM.

VARPTR (# numarq)

Função que fornece o endereço de memória do início do buffer do arquivo especificado.

numarq = número do arquivo aberto cujo endereço do início do buffer se quer determinar.

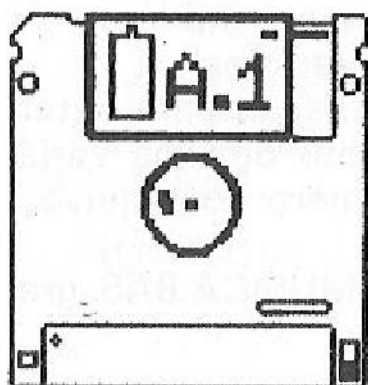
A função VARPTR pode também ser usada para determinar o endereço inicial do armazenamento de uma variável (numérica ou string) no lugar do número do arquivo.

Exemplo:

Inclua a linha a seguir no programa MUSICA.BAS gravado no disco mestre e veja seu efeito.

```
1155 LOCATE 5,20:PRINT "O Buffer
     comeca em";HEX$(VARPTR(#1))
```

VERIFY

Veja CALL VERIFY.

MENSAGENS DE ERROS

A seguir estão relacionadas as mensagens de erro exclusivas do DISK BASIC (gravadas na interface) e do MSXDOS (gravadas nos arquivos MSXDOS.SYS e COMMAND.COM). Lembre-se que quando você está trabalhando em DISK BASIC as mensagens cujos códigos estão entre 0 e 60 são as mesmas do BASIC comum. Para saber o que elas significam, consulte o manual de seu microcomputador.

MENSAGENS DO DISK BASIC MSX

BAD FAT **ERRO 60**

O disco acessado não está com a FAT correta. A FAT (File Allocation Table) fica nos setores 1, 2, 3 e 4 do disco e armazena os setores ocupados por cada arquivo. Se você não conseguir acessar nenhum arquivo, a solução será reformatar o disco.

BAD FILE MODE **ERRO 61**

O arquivo que se tentou acessar foi aberto de forma incorreta. Abra o arquivo corretamente ou use a forma de acesso correta.

BAD DRIVE NAME **ERRO 62**

O drive especificado não está conectado ao sistema. Use outro drive.

BAD SECTOR NUMBER **ERRO 63**

O setor que o sistema tentou acessar não existe.

FILE STILL OPEN **ERRO 64**

O arquivo que se tentou abrir, já está aberto.

FILE ALREADY EXISTS **ERRO 65**

Você tentou renomear um arquivo com um nome de arquivo que existe no disco. Use outro nome.

DISK FULL **ERRO 66**

O disco está cheio e não permite a gravação de mais informações. Use outro disco ou apague algum arquivo que não será utilizado.

TOO MANY FILES **ERRO 67**

O diretório do disco não suporta mais arquivos (cabem no máximo 112). Apague algum arquivo inútil ou use outro disco.

DISK WRITE PROTECT **ERRO 68**

O disco em que se tentou gravar algum dado está protegido contra gravação. Retire o disco do drive e destrave-o.

DISK I/O ERROR **ERRO 69**

O disco está com algum dano físico irrecuperável ou os setores que definem a formatação estão danificados.

DISK OFFLINE **ERRO 70**

O disco que se tentou acessar não está corretamente inserido no drive. Retire e torne a inserir o disco no drive. Se o problema persistir, seu drive pode estar desalinhado e você deve levá-lo a uma assistência técnica.

RENAME ACROSS DISK **ERRO 71**

O comando NAME foi usado com os nomes dos dispositivos incorretos. O drive especificado junto com o novo nome difere do especificado junto ao nome antigo. Use o mesmo drive nos dois parâmetros!

MENSAGENS DO MSXDOS

ABORT, RETRY, IGNORE?

Algum comando não pode ser executado. Você deve digitar A se quiser desistir do comando, R se quiser reentrá-lo e I se quiser que o sistema ignore o erro detectado.

ARE YOU SURE (Y/N)?

Esta mensagem surge quando se comanda DEL *.* ou ERASE *.* e permite um arrependimento.

BAD COMMAND OR FILE NAME

Você digitou um comando que não existe no MSXDOS e que não é nome de nenhum arquivo ".COM" ou ".BAT". Corrija o comando ou especifique o drive em que está o arquivo a ser executado.

BAD FAT, DRIVE X

A FAT (Tabela de Alocação de Arquivos) do disco presente no drive X não pôde ser lida pelo sistema.

CONTENT OF DESTINATION LOST BEFORE COPY

O comando COPY foi usado com um arquivo de destino já existente em disco. O antigo conteúdo desse arquivo foi perdido, sendo substituído pelo que lhe foi atribuído com o COPY.

DATA ERRADA
ENTRE NOVA DATA

Você entrou com uma data em formato inválido. Entre com a data no formato correto ou digite RETURN.

DISK ERROR READING DRIVE X ou
DISK ERROR WRITING DRIVE X

O disco está com algum dano físico irrecuperável ou os setores que definem a formatação estão danificados logicamente.

FILE CANNOT BE COPIED ONTO ITSELF

O comando COPY foi usado para uma cópia de um arquivo sobre ele mesmo. Use um nome ou extensão diferente do arquivo original para o arquivo cópia.

FILE CREATION ERROR

O MSXDOS não conseguiu criar um arquivo.

FILE NOT FOUND

Um arquivo fornecido como parâmetro no último comando digitado, não existe no drive especificado (ou no drive corrente). Verifique se o nome fornecido existe em algum dos drives conectados ao sistema e corrija o comando.

HORA ERRADA
ENTRE NOVA HORA

Você entrou com uma hora não interpretável pelo MSXDOS. Entre a hora no formato correto ou digite RETURN.

INSERT DISK WHIT BATCH FILE
AND STRIKE ANY KEY WHEN READY

Este erro acontece quando, durante a execução de um arquivo BATCH, o disco que o contém é retirado do drive e substituído por outro.

INSERT DOS DISK IN DEFAULT DRIVE
AND STRIKE ANY KEY WHEN READY

Esta mensagem é apresentada quando o sistema procura o arquivo COMMAND.COM no disco e não o acha. Insira um disco com o MSXDOS e com o COMMAND.COM no drive acessado.

INSUFFICIENT DISK SPACE

O disco em que se tentou gravar informações não dispõe de espaço suficiente. Use outro disco ou elimine algum arquivo inútil do disco acessado.

INVALID DRIVE SPECIFICATION

Foi especificado um drive não conectado ao sistema. Use outro drive.

INVALID PARAMETER

Foi fornecido um parâmetro inválido junto a um comando (MODE 0, por exemplo). Use um parâmetro correto.

NOT READY

O disco que o sistema tentou acessar não está corretamente formatado. Formate-o ou troque de disco.

PROGRAM TOO BIG TO FIT IN MEMORY

O arquivo a ser carregado para a memória do micro é maior que a área de memória disponível

RENAME ERROR

Você tentou renomear um arquivo com um nome já existente no disco. Use outro nome.

STRIKE A KEY WHEN READY

Você deve pressionar alguma tecla para que o sistema prossiga.

TERMINATE BATCH FILE (Y/N)?

A execução de um arquivo BATCH foi interrompida por ^C ou ^STOP e o sistema pede confirmação para cancelá-lo em definitivo.

UNSUPPORTED MEDIA TYPE

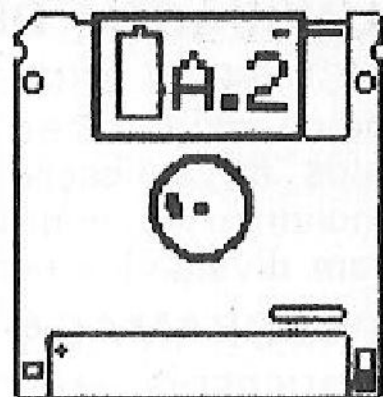
O disco usado está formatado de modo incompatível com o MSXDOS.

WRITE ERROR

Ao tentar gravar dados o MSXDOS detectou algum erro. Tente reentrar o comando. Se o erro persistir, reformate o disco e tente usar o comando mais uma vez. Se ainda assim o erro persistir, provavelmente seu disco está com danos físicos irrecuperáveis em alguns de seus setores.

WRITE PROTECT

Foi feita uma tentativa de gravar algum dado num disco protegido contra gravação. Retire o disco do drive, destrave-o, torne a inserí-lo e reentre com o comando.

BIBLIOGRAFIA ACONSELHADA

Para tirar o máximo proveito do seu disk drive LEOPARD para MSX é conveniente que você tenha um bom conhecimento a respeito do próprio microcomputador. Para isso, além dos manuais que acompanham o micro, aconselhamos a leitura de algumas obras editadas pela ALEPH para que você possa se aprofundar no conhecimento do seu MSX.

CURSO DE BASIC MSX — vol. 1

CARVALHO JR. & PIAZZI — Neste livro você tem uma introdução clara e didática ao BASIC residente do MSX. Esta obra é muito importante para os que estão se iniciando no mundo da computação.

COLEÇÃO DE PROGRAMAS — vol. 1

OLIVEIRA & OUTROS — Neste volume é apresentada uma coletânea de programas de fácil digitação que permitem, através dos comentários e descrição, uma rápida familiarização com a linguagem BASIC.

COLEÇÃO DE PROGRAMAS — vol. 2

OLIVEIRA & OUTROS — Trata-se da continuação da coletânea do volume 1 com exemplos de programas didáticos, utilitários e aplicativos. Para os possuidores de drive há um programa muito útil que permite ler o cabeçalho de programas em linguagem de máquina gravados em fita para se descobrir o endereço de início, fim e execução. Estes dados são indispensáveis para posterior gravação em disco com BSAVE.

LINGUAGEM DE MÁQUINA PARA MSX

FIGUEREDO & ROSSINI — O ABC da linguagem de máquina do microprocessador Z-80 (utilizado no MSX) com exemplos de aplicação no MSX. Obra exaustiva e completa, incluindo as instruções secretas do Z-80 (as que não foram divulgadas nem pelo fabricante do chip)!

PROGRAMAÇÃO AVANÇADA EM MSX

FIGUEREDO, MALDONADO & ROSSETTO — Como tirar o máximo proveito de seu MSX usando o ASSEMBLY como linguagem de programação. Rico em exemplos e rotinas úteis.

APROFUNDANDO-SE NO MSX

PIAZZI, MALDONADO, OLIVEIRA & OUTROS — O "best seller" da literatura sobre MSX. Neste livro, dedicado a leitores que tenham pouca ou até nenhuma familiaridade com a Linguagem de Máquina, toda a arquitetura do MSX é dissecada com explicações didáticas e exemplos claros e úteis. Este livro foi considerado por milhares de leitores como a "bíblia" do MSX!

Se você está buscando informações sobre outras interfaces e sistemas operacionais, consulte mais dois livros editados pela ALEPH:

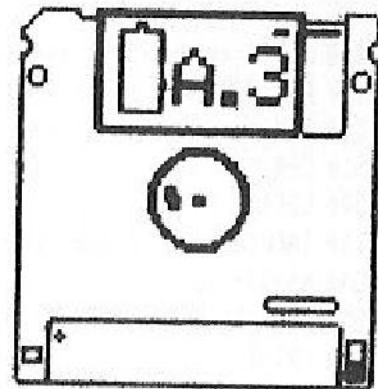
SISTEMA DE DISCO PARA MSX

OLIVEIRA & PEREIRA

USANDO O DISK DRIVE NO MSX

PEREIRA

Se você quiser receber gratuitamente o boletim informativo da ALEPH, contendo programas e dicas para seu MSX e todas as novidades sobre a literatura produzida por nossa editora, envie seu nome e endereço completos (inclusive CEP) para EDITORA ALEPH — C.P. 20.707 — CEP 01498 — S. Paulo — SP.

EXEMPLOS DE PROGRAMAS

Para compreender melhor o uso de vários comandos e funções do DISK BASIC, listamos a seguir 3 programas-exemplo:

ESTATIST.BAS

```
1000 REM ==============================
1010 REM ====   EXEMPLO DE USO    ====
1020 REM === DE ARQUIVO SEQUENCIAL ===
1030 REM ==============================
1040 COLOR 15,1,1 : SCREEN 0
1050 KEY OFF : POKE &HFCAB,1
1060 DIM A(1000)
1070 LOCATE 10,0 : PRINT"ESTATISTICA"
1080 LOCATE 10,1 : PRINT"==========="
1090 LOCATE 5,5
1100 PRINT"[1] LER TABELA DE DADOS"
1110 LOCATE 5,7
1120 PRINT"[2] ACRESCENTAR DADOS"
1130 LOCATE 5,9
1140 PRINT"[3] GERAR NOVA TABELA"
1150 LOCATE 5,11 : PRINT"[4] FIM"
1160 LOCATE 3,15
1170 PRINT"QUAL A SUA ESCOLHA ? ";
1180 Q$=INPUT$(1)
1190 IF Q$="1" THEN 1270
1200 IF Q$="2" THEN 1420
1210 IF Q$="3" THEN 1510
1220 IF Q$="4" THEN 2190
1230 BEEP : BEEP : GOTO 1180
1240 REM ==============================
1250 REM =====  LEITURA DE DADOS ======
1260 REM ==============================
1270 CLS
1280 LOCATE ,5
1290 INPUT"TABELA A SER LIDA:";N$
1300 N$=N$+".DAT"
1310 OPEN N$ FOR INPUT AS #1
1320 LINE INPUT #1,A$
1330 N=1
1340 IF EOF(1)=-1 THEN 1380
1350 INPUT #1,A(N)
1360 N=N+1
1370 GOTO 1340
1380 N=N-1:CLOSE #1:GOTO 1770
1390 REM ==============================
1400 REM === ACRESCIMO DE REGISTROS ===
1410 REM ==============================
1420 CLS
1430 LOCATE ,5
1440 INPUT"TABELA A SER ACRESCIDA ";N$
1450 N$=N$+".DAT"
1460 OPEN N$ FOR APPEND AS #1
1470 GOTO 1600
```

```
1480 REM ================================
1490 REM =ENTRADA E GRAVACAO DOS DADOS=
1500 REM ================================
1510 CLS
1520 LOCATE ,5
1530 INPUT"TABELA A SER GERADA ";N$
1540 N$=N$+".DAT"
1550 OPEN N$ FOR OUTPUT AS #1
1560 PRINT
1570 PRINT"ENTRE O TITULO DA TABELA"
1580 LINE INPUT A$
1590 PRINT #1,A$
1600 CLS
1610 N=1
1620 PRINT"ENTRE OS DADOS: (F=FIM)"
1630 PRINT
1640 PRINT"DADO";N;:INPUT X$
1650 IF X$="F" THEN 1700
1660 A(N)=VAL(X$)
1670 PRINT #1,A(N)
1680 N=N+1
1690 GOTO 1640
1700 CLOSE #1
1710 CLS
1720 GOTO 1070
1730 REM ================================
1740 REM === ROTINA DE APRESENTACAO ===
1750 REM ====  DOS DADOS E CALCULO  ===
1760 REM ================================
1770 CLS
1780 PRINT"APRESENTACAO DOS DADOS";
1790 PRINT" DA TABELA:"
1800 PRINT A$:PRINT
1810 FOR I=1 TO N
1820 PRINT"DADO";I;
1830 LOCATE 15:PRINT A(I)
1840 IF I MOD 20 = 0 THEN Q$=INPUT$(1)
1850 NEXT I
1860 PRINT : PRINT"CALCULANDO ..."
1870 S=0 : FOR I=1 TO N
1880 IF I=1 THEN MA=A(I):MI=MA
1890 IF MI>A(I) THEN MI=A(I)
1900 IF MA<A(I) THEN MA=A(I)
1910 S=S+A(I):NEXT I
1920 ME=S/N
1930 DM=0 : VA=0 : S=0
1940 FOR I=1 TO N
1950 DM=DM+ABS(A(I)-ME)
1960 S=A(I)-ME
1970 VA=VA+S^2
1980 NEXT
1990 PRINT : PRINT"APERTE UMA TECLA";
2000 PRINT" PARA OS RESULTADOS";
2010 Q$=INPUT$(1) : PRINT : PRINT
2020 PRINT STRING$(40,195);
2030 PRINT"VALOR MINIMO  =";
2040 LOCATE 15 : PRINT MI
2050 PRINT"VALOR MAXIMO  =";
2060 LOCATE 15 : PRINT MA
2070 PRINT"MEDIA         =";
2080 LOCATE 15 : PRINT ME
2090 PRINT"DESVIO MEDIO  =";
2100 LOCATE 15 : PRINT DM/N
2110 PRINT"VARIANCIA     =";
2120 LOCATE 15 : PRINT VA/N
2130 PRINT"DESVIO PADRAO =";
2140 LOCATE 15 : PRINT SQR(VA/N)
2150 PRINT STRING$(40,192);
2160 Q$=INPUT$(1)
2170 CLS
2180 GOTO 1070
2190 POKE &HFCAB,0 : CLS : END
```

MUSICA:BAS

```
1000 REM ===============================
1010 REM ===== EXEMPLO DE USO DE =====
1020 REM ==== ARQUIVOS RANDOMICOS ====
1030 REM ===============================
1040 OPEN"MUSICA.DAT" AS #1 LEN=89
1050 FIELD #1,30 AS M$,20 AS D$,20 AS C$
          , 4 AS T$,15 AS G$
1060 REM ===============================
1070 REM ========    MENU    =========
1080 REM ===============================
1090 COLOR 15,1,1 : SCREEN 0
1100 KEY OFF : POKE &HFCAB,1
1110 LOCATE8:PRINT"ARQUIVO DE MUSICAS"
1120 LOCATE10,5:PRINT"[I] INSERIR"
```

```
1130 LOCATE10,7:PRINT"[L] LER"
1140 LOCATE10,9:PRINT"[A] ALTERAR"
1150 LOCATE10,11:PRINT"[F] FIM"
1160 LOCATE8,15:PRINT"QUAL SUA OPCAO?";
1170 A$=INPUT$(1)
1180 IF A$="I" THEN 1260
1190 IF A$="L" THEN 1570
1200 IF A$="A" THEN 1970
1210 IF A$="F" THEN 2090
1220 BEEP : BEEP : GOTO 1170
1230 REM ==============================
1240 REM =====  ENTRADA DOS DADOS =====
1250 REM ==============================
1260 RE=LOF(1)/89+1
1270 CLS
1280 PRINT"REGISTRO";RE:PRINT:PRINT
1290 INPUT"MUSICA ";MU$:PRINT
1300 INPUT"CONJUNTO ";CJ$:PRINT
1310 INPUT"DISCO ";DI$:PRINT
1320 INPUT"TEMPO (MMSS) ";TE$:PRINT
1330 INPUT"GENERO ";GE$
1340 PRINT:PRINT
1350 PRINT"DADOS CORRETOS ? (S/N) ";
1360 A$=INPUT$(1)
1370 IF A$="S" THEN 1430
1380 IF A$="N" THEN 1270
1390 BEEP : BEEP : GOTO 1360
1400 REM ==============================
1410 REM ===== GRAVACAO DOS DADOS =====
1420 REM ==============================
1430 LSET M$=MU$ : LSET C$=CJ$
1440 LSET D$=DI$ : LSET T$=TE$
1450 LSET G$=GE$
1460 PUT #1,RE
1470 IF FL=1 THEN RETURN
1480 PRINT : PRINT
1490 PRINT"QUER INSERIR OUTRO? (S/N) ";
1500 A$=INPUT$(1)
1510 IF A$="S" THEN RE=RE+1 : GOTO 1270
1520 IF A$="N" THEN 1090
1530 BEEP : BEEP : GOTO 1500
1540 REM ==============================
1550 REM = LEITURA E SELECAO DOS DADOS=
1560 REM ==============================
1570 CLS
1580 LOCATE 10:PRINT"LEITURA DOS DADOS"
1590 LOCATE 8,5
1600 PRINT"[T] TODOS OS REGISTROS"
1610 LOCATE 8,7
1620 PRINT"[M] SELECAO POR MUSICA"
1630 LOCATE 8,9
1640 PRINT"[G] SELECAO POR GENERO"
1650 LOCATE 8,11
1660 PRINT"[C] SELECAO POR CONJUNTO"
1670 LOCATE 10,15
1680 PRINT"QUAL SUA ESCOLHA ? ";
1690 A$=""
1700 A$=INPUT$(1)
1710 IF A$<>"T" AND A$<>"M" AND A$<>"G"
AND A$<>"C" THEN BEEP : BEEP : GOTO 1700
1720 PRINT : PRINT
1730 IF A$="M" THEN PRINT : INPUT"QUAL M
USICA ";MU$
1740 IF A$="G" THEN PRINT : INPUT"QUAL G
ENERO ";GE$
1750 IF A$="C" THEN PRINT : INPUT"QUAL C
ONJUNTO ";CJ$
1760 CLS
1770 FI=LOF(1)/89:IF FI=0 THEN 1090
1780 FOR I=1 TO FI
1790 GET #1,I
1800 IF A$="M" AND MU$<>LEFT$(M$,LEN(MU$
)) THEN 1920
1810 IF A$="C" AND CJ$<>LEFT$(C$,LEN(CJ$
)) THEN 1920
1820 IF A$="G" AND GE$<>LEFT$(G$,LEN(GE$
)) THEN 1920
1830 PRINT"REGISTRO";I
1840 PRINT:PRINT
1850 PRINT"MUSICA : ";M$:PRINT
1860 PRINT"CONJUNTO : ";C$:PRINT
1870 PRINT"DISCO : ";D$:PRINT
1880 PRINT"TEMPO : ";T$:PRINT
1890 PRINT"GENERO : ";G$:PRINT
1900 PRINT:PRINT:IF FL=1 THEN RETURN
1910 Q$=INPUT$(1)
1920 NEXT I
1930 GOTO 1090
1940 REM ==============================
1950 REM ==== ROTINA DE ALTERACAO =====
1960 REM ==============================
1970 LOCATE 10,17
1980 INPUT"QUAL REGISTRO ";RE:FL=1
1990 GET #1,RE : CLS : PRINT"REGISTRO";R
E
```

```
2000 CLS : PRINT"REGISTRO";RE
2010 PRINT : PRINT:GOSUB 1050
2020 PRINT"CONFIRMA ? (S/N) ";
2030 Q$=INPUT$(1)
2040 IF Q$="S" THEN LOCATE 0,15 : PRINT
SPC(16) : LOCATE 0,15 : GOTO 2070
2050 IF Q$="N" THEN 2080
2060 BEEP : BEEP : GOTO 2030
2070 GOSUB 1290
2080 FL=0 : GOTO 1090
2090 POKE &HFCAB,0 : CLS : END
```

LISTPRE.BAS

```
100 REM ================================
110 REM EXEMPLO DE USO DE CVI,CVS e CVD
120 REM        e MKI$, MKS$, MKD$
121 REM ================================
130 KEYOFF : CLS : POKE &HFCAB,1
140 OPEN "LISTPRE.DAT" AS #1 LEN=24
150 FIELD #1,10 AS  N$,2 AS  Q$,
            4 AS PB$,8 AS PL$
160 LOCATE 5:PRINT"LISTA DE PRECOS"
170 LOCATE 5,5
171 PRINT"[A] GRAVAR REGISTROS"
180 LOCATE 5,7
181 PRINT"[B] LER OU ALTERAR REGISTROS"
190 LOCATE 5,9:PRINT"[C] FIM"
200 LOCATE 2,13
201 PRINT"QUAL SUA ESCOLHA ? ";
210 H$=INPUT$(1)
220 IF H$="A" THEN 260
230 IF H$="B" THEN 450
240 IF H$="C" THEN 2000
250 BEEP : BEEP : GOTO 210
260 REM ================================
270 REM ========== GRAVACAO ===========
280 REM ================================
290 CLS
300 R=LOF(1)/24+1
310 PRINT"REGISTRO";R : PRINT
320 INPUT"MERCADORIA ";M$
330 IF M$="FIM" THEN CLS:GOTO 160
340 INPUT"QUANTIDADE ";Q%
350 INPUT"PRECO UNITARIO BRUTO ";PB!
360 INPUT"DESCONTO (%) ";D
370 PL#=(PB!)*((100-D)/100)
380 LSET N$=M$
390 RSET Q$=MKI$(Q%)
400 RSET PB$=MKS$(PB!)
410 RSET PL$=MKD$(PL#)
420 PUT #1,R
430 IF FL=1 THEN FL=0:RETURN
440 R=R+1 : PRINT : GOTO 310
450 REM ================================
460 REM ===== LEITURA E ALTERACAO =====
470 REM ================================
480 CLS
490 FOR I=1 TO LOF(1)/24
500 GET #1,I
510 Q%=CVI(Q$)
520 PB!=CVS(PB$)
530 PL#=CVD(PL$)
540 PRINT"REGISTRO";I : PRINT
550 PRINT"MERCADORIA         = ";N$
560 PRINT"QUANTIDADE        =";Q%
570 PRINT"PRECO UNIT.BRUTO =";PB!
580 PRINT"PRECO UNIT.LIQ.  =";PL#
590 PRINT:PRINT"QUER ALTERAR ? (S/N) ";
600 H$=INPUT$(1)
610 IF H$="S" THEN :R=LOC(1): CLS : FL=1
 : GOTO 640
620 IF H$="N" THEN GOTO 650
630 BEEP : BEEP : GOTO 600
640 GOSUB 310
650 PRINT : PRINT
660 NEXT I
670 CLS : GOTO 160
2000 POKE &HFCAB,0 : CLS : END
```

DRIVES

NOVOS HORIZONTES PARA O SEU MSX

Este livro destina-se aos usuários de MSX que desejam conhecer os novos horizontes que se abrem para seu micro quando conectado a um acionador de discos. Ele será extremamente útil aos que já possuem um disk drive de 3 1/2" ou 5 1/4" e desejam tirar o máximo proveito de seu equipamento.

Estruturado de maneira tutorial, ele permite aprender, passo-a-passo, tudo a respeito do MSXDOS e do DISK BASIC.

No DISK BASIC, o leitor aprende a criar e gerenciar arquivos sequenciais e randômicos. Para melhor compreensão, temos um apêndice com 3 programas-exemplo que utilizam todas as funções e comandos mais importantes desta linguagem.

No MSXDOS, o usuário pode descobrir como criar arquivos BATCH, inclusive para auto-execução.

Para consulta, existem dois dicionários de comandos e funções, tanto do MSXDOS quanto do DISK BASIC.

Trata-se de uma obra indispensável para quem deseja operar, com conhecimento de causa, um acionador de disco conectado a um microcomputador padrão MSX.